



TITLE:

情報基礎演習 2017

AUTHOR(S):

喜多, 一; 北村, 由美; 日置, 尋久

CITATION:

喜多, 一 ...[et al]. 情報基礎演習 2017. 2017: 1-153

ISSUE DATE:

2017-03-01

URL:

<http://hdl.handle.net/2433/224813>

RIGHT:

本書はCC-BY-NC-ND ライセンスによって許諾されています。ライセンスの内容を知りたい方は<https://creativecommons.org/licenses/by-nc-nd/4.0/deed.ja> でご確認ください。

情報基礎演習 2017

京都大学 国際高等教育院 喜多 一
附属図書館 北村由美
人間・環境学研究科 日置尋久

Version 2017/03/21

0. まえがき

京都大学では全学共通科目として情報基礎演習が実施されています。この科目で扱う内容は学部、学科ごとに専門での必要性に応じて異なりますが、この教科書は2017年度の情報基礎演習（全学向）での利用を想定して編集されています。

本書は1章でも述べているように、大学での学習にICTを利用するためのスキル、すなわちアカデミックな活動のためのICTスキルの獲得を主な目的としています。授業では紹介しないものも自学自習に役立てて頂くために含めています。演習も多く示していますが、実際の授業では取捨選択して取り組みます。

授業は京都大学の情報環境機構が提供する教育用コンピュータシステムの端末を用いて行うため、パーソナルコンピュータ環境はオペレーティングシステムとしてWindows7を、またオフィスソフトとしてMicrosoft Office 2010を想定しています。13章ではプログラミングを取り上げていますが、プログラミング言語としては初学者にも学びやすく、学術的に高度な利用も進んでいるPythonを取り上げました。また、学生の皆さんの学習成果の確認のための科目内容の達成水準を示した評価基準（ルーブリック）を付録としてつけています。

現在、私たちが使っている情報通信技術は数多くの人々の努力により構築されているものです。このようにして技術が構築されてきたことと先人の努力に感謝する意味で、コラムとして「Great Idea」を設けて紹介しています。

執筆は4章を北村が、13章を日置が、他の章は喜多が担当しました。本書が皆さんのICTスキル向上に役立てば幸いです。

なお取り上げた内容は国際高等教育院に設けられた情報科目設計ワーキングチームでまとめられた演習系科目の構成に準じて選定しています。また4章の執筆にあたっては京都大学附属図書館利用支援課、原竹留美氏、櫻井待子氏、坂本拓氏にご協力いただきました。ご協力いただいた皆様に心より御礼申し上げます。

本書はCC-BY-NC-NDライセンスによって許諾されています。ライセンスの内容を知りたい方は <https://creativecommons.org/licenses/by-nc-nd/4.0/deed.ja> でご確認ください。



目次

0. まえがき	2
1. この授業の目的, 評価基準, 進め方	8
1.1 この授業の3つの目的	8
1.2 評価基準	10
1.3 授業の進め方	10
1.4 「効果」と「効率」からの評価と改善	11
1.5 この教科書の構成	11
2. コンピュータの基礎	2
2.1 学習の目的	2
2.2 情報通信技術の進展	2
2.3 パーソナルコンピュータの構成	5
Great Idea! ストアードプログラム方式	6
2.4 オペレーティングシステムの役割	8
Great Idea! オペレーティングシステム	9
Great Idea! Window, Icon, Menu & Pointer	12
2.5 自立したパソコン利用者に求められるスキル	12
2.6 キーボードのタイピング	15
Great Idea! かな漢字変換	20
2.7 ソフトウェアのライセンスと保守	21
参考文献	25
3. ネットワークの基礎	26
3.1 学習の目的	26
3.2 仮想社会の安全運転	26
3.3 情報セキュリティ e-Learning の受講	27
3.4 ローカルエリアネットワーク (LAN) の基礎	27
3.5 インターネットの基礎	30
Great Idea! インターネットプロトコル	32
Great Idea! 公開鍵暗号	38

3.6	パーソナルコンピュータのネットワーク接続	39
3.7	電子メールの利用	40
3.8	情報通信技術 (ICT) の利用とリスク	42
4.	知的生産について.....	45
4.1	本章の目的	45
4.2	知的生産とは	45
4.3	学園祭の活動と知的生産.....	46
4.4	大学生活と知的生産	47
4.5	知的生産を構成する活動.....	48
4.6	知的生産と ICT 活用.....	49
4.7	知的生産スキルの獲得と学習の方略.....	50
	参考文献	51
5.	学術情報の探索	52
5.1	本章の目的	52
5.2	学術情報と図書館	52
5.3	検索を行う前に	55
5.4	参考文献リストの読み方・書き方.....	58
5.5	検索の実際	62
5.6	困ったら図書館に	62
	参考文献	63
	Great Idea! ガーフィールド博士と引用文献データベース	63
6.	表計算ソフトによるデータ処理 (1)	64
6.1	学習の目的	64
6.2	表計算ソフトの利用目的.....	64
6.3	表計算ソフトの基本構造.....	64
	Great Idea! 表計算ソフトの開発	67
6.4	データの入力方法.....	68
6.5	データの並べ替え	73
6.6	データの集計	74

6.7	データのクロス集計	76
7.	表計算ソフトによるデータ処理（2）	79
7.1	学習の目的	79
7.2	シミュレーションとは	79
7.3	動的な状況のシミュレーションの構成要素	79
7.4	借入金の返済のシミュレーション	79
7.5	バネ・ダンパモデルのシミュレーション	82
8.	表計算ソフトによるデータ処理（3）	87
8.1	学習の目的	87
8.2	データをグラフで表示することの意味	87
8.3	データをグラフで表現する時の注意事項	87
8.4	グラフを Excel で作成する際の用語	88
8.5	散布図を Excel で描く	89
8.6	棒グラフや折れ線グラフを Excel で描く	94
8.7	ヒストグラムを描く	95
9.	構造化された文書の作成	98
9.1	学習の目的	98
9.2	論文やレポートの構造	98
9.3	人がすること，コンピュータに任せること	99
9.4	スタイルとテンプレート	101
9.5	パラグラフの書き方	103
9.6	章立ての見出しの書き方	104
9.7	アウトライン表示の活用	105
9.8	参考文献の記載方法とその引用	106
9.9	文章の校閲方法	107
9.10	PDF 形式の文書の作成	108
	参考文献	108
10.	文章への図表の挿入	110

10.1	学習の目的	110
10.2	学術的文章での図表番号，タイトル，説明の付け方	110
10.3	Word での図表番号の付与と相互参照	111
10.4	Word の文章への図表の挿入	111
10.5	Word と Excel, PowerPoint の連携	112
10.6	PowerPoint の図の挿入	113
10.7	Excel の表やグラフの挿入	114
11.	文書での数式の扱い.....	115
11.1	学習の目的	115
11.2	表現の手段としての数式.....	115
11.3	変数は傾いている	115
11.4	Word での数式の書き方.....	116
11.5	2次元形式と行形式.....	117
12.	プレゼンテーションについて考えよう	118
12.1	学習の目的	118
12.2	プレゼンテーションの目的と聞き手	118
12.3	プレゼンテーションでの留意事項.....	118
12.4	スライド作成の基本	119
12.5	発表の作法	123
12.6	グループでのプレゼンテーション	125
12.7	聞き手としての態度	125
12.8	プレゼンテーションの評価	125
12.9	ポスター発表について	126
	参考文献	128
13.	プログラミングの基礎	129
13.1	学習の準備	129
13.2	学習の目的	129
13.3	プログラムとプログラミングの概要	130
	Great Idea! 自動プログラミング	138

0 まえがき

13.4	変数によるモデル化	139
13.5	制御構造 — 条件に基づく処理	143
13.6	プログラミング演習の準備	146
	参考文献	151
14.	あとがき	152
別表 1	評価基準	153

1. この授業の目的， 評価基準， 進め方

1.1 この授業の 3 つの目的

大学での学習では文献などの情報探索を行ったり，統計データを活用したりしつつレポート作成し，プレゼンテーションを行うなど様々な場面で ICT の活用が求められます。すなわちアカデミックな活動に必要な ICT スキルを身に着けることが求められます。また，大学生になると，クラブやサークル，アルバイトや NPO での活動など授業外の活動も拡大し，より主体的に活動を実施するようになります。このような活動においてもフォーマルな場面で ICT を活用するスキルが求められます。これに必要なスキルはアカデミックな活動のための ICT スキルと近いものです。この授業の第一の目的はアカデミックな活動に必要な ICT スキルを身に着けることです。

次のポイントは大学に入学して一人暮らしを始める学生も多くなりすし，ネットワークに接続されたパーソナルコンピュータ（PC）を個人で占有利用するようになることです。このため PC を適切に運用することなど，自立した ICT ユーザとしてのスキルも必要になります。これがこの授業の第二の目的です。

大学でのアカデミックな活動は学年進行とともに内容が高度化，専門化し，これに伴ってより高いレベルでのスキルが求められます。しかしながら，これを 1 年生の限られた時間で学習することは難しいですし，みなさん自身にとっても必ずしも利用するとは限らない高度な内容を学習することの意義を見いだせないかもしれません。しかしながら，2 年生以降，まとまって ICT スキルを学ぶ授業などが設けられているとは限りませんので自身で主体的に学ぶことが求められます。この授業の第三の目的は，学年進行に伴って高度化する学術的な活動のための ICT スキルを自主的・継続的に獲得するための自学自習能力を身に着けることです。

これらのことをまとめるとこの授業の目的は以下のようになります。

1 この授業の目的，評価基準，進め方

- アカデミックな活動に必要な ICT スキルを獲得すること：大学が提供するさまざまな情報サービスを適切に利用できるようなるとともに，アカデミックな活動に必要な情報探索，データの処理，レポート作成とプレゼンテーションなどのための ICT スキル獲得することです。
- 自立した ICT ユーザとなること：パーソナルコンピュータやネットワークを自立したユーザとして適切に利用し管理する能力を身に着けることです。
- ICT スキルを自主的・継続的に獲得する自学自習能力を身につけること：この授業では扱いませんが，学年が進行し，より専門的な能力を身に着けて行く上で必要なスキルを主体的に獲得することです。

科目は半期1コマで1年生の最初の学期で実施されることを想定しています。これら全体を通して学習の目的を表現するならば図2のように大学での学習の生産性を早い段階で高め，大学での学習をより成功させるための ICT スキルの獲得と言えるでしょう。

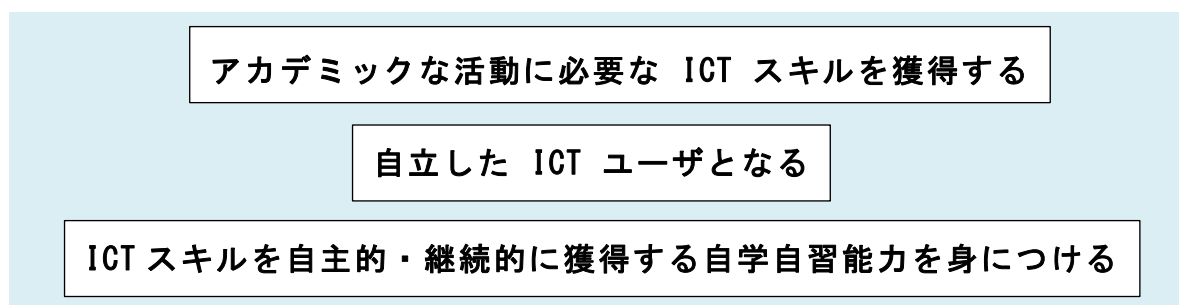


図 1 この科目の3つの目的

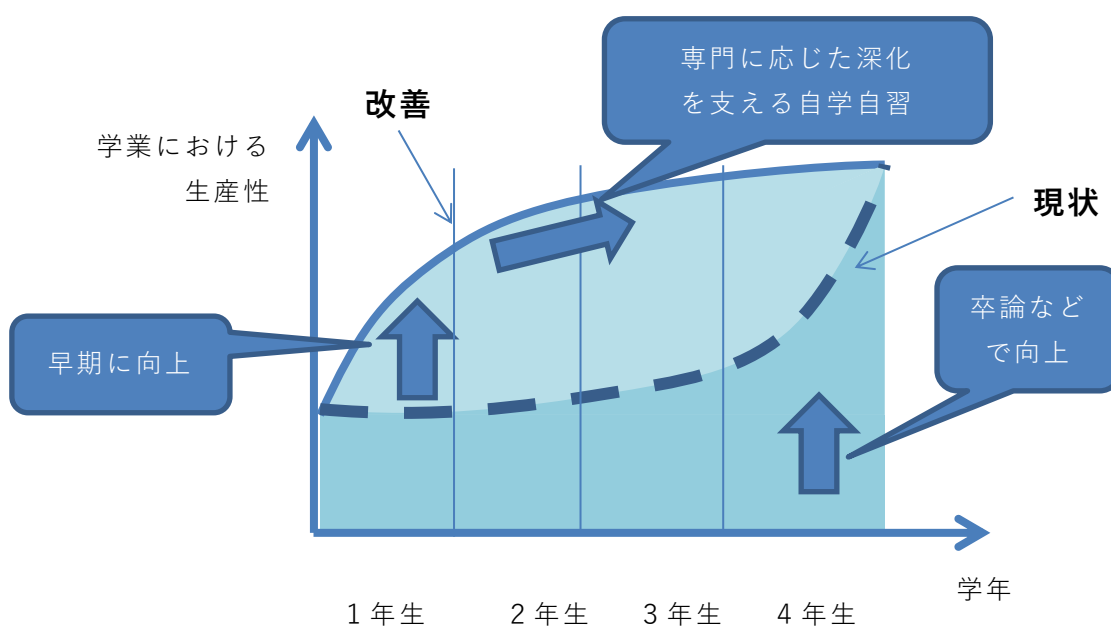


図 2 学業における生産性の向上

1.2 評価基準

それでは、どのような能力を身につければいいのでしょうか。別表 1 は学習する事項を縦軸に，達成すべきレベルを横軸に取り，各マス目にその事項の求められる達成レベルを示したものです。レベル C は最低限，達成して欲しいレベル，レベル A はこの授業で（あるいは 1 年生で）目標とすべきレベル，レベル S は卒業時までには身につけてほしいレベルとして設けています。

演習 1. ICT スキルの自己評価

別表 1 の各項目について，ご自身の現在のスキルレベルを確認し，達成すべき目標との差を理解してください。

1.3 授業の進め方

大学での授業は授業時間内と授業時間外の学習で構成されます。授業の進め方は担当教員によって異なりますが，本書では演習課題の指示に沿って各自で演習を進め，課題を授業支援システム PandA で提出する形で進

めることを想定しています。授業中に終了しなかった課題は授業後に自習し PandA で提出することも想定しています。このほか，授業時間外に行う学習課題も課せられますが，これも PandA で提出することが行われるでしょう。

1.4 「効果」と「効率」からの評価と改善

スキルを身に着けて行く上での視点として「効果」と「効率」という 2 つの視点をもって自身の活動を自分で評価してください。

- **効果的 (Effective)** とは実現したい目的により適した方法が取られていて，よりよい成果が得られていることを言います。これに対して
- **効率的 (Efficient)** とは同じことを実現するのにより早く，より正確に，より少ない手間でできる方法を用いていることを言います。

ご自身の活動をこの 2 つの視点で常に自己評価し，より効果的で効率的に活動できるように

- 継続的に学習し作業を改善すること
- を心がけてください。大量の情報を高速で扱えるコンピュータの特性から，PC など情報機器やサービスの利用ではこのような継続的な学習と改善が極めて重要です。

1.5 この教科書の構成

この教科書は授業で取り組む内容について，学習の目標を示し，学習する内容について解説します。演習問題をいくつか設定していますが授業で実施する演習課題については別途，具体的に指示されます。

2. コンピュータの基礎

2.1 学習の目的

この章では自立的なコンピュータの利用者となるため、

- 自立したパーソナルコンピュータの利用者に求められるスキル

について説明したあと、これを体系的に理解するために

- パーソナルコンピュータの構成と
- オペレーティングシステム役割

について学びます。

また、パーソナルコンピュータの操作の中でも、アカデミックな活動をより効率良く行うための基礎的なスキルとして

- キーボードのタイピング

について学びます。

2.2 情報通信技術の進展

情報通信技術がどのように進展してきたかの概要を知ることは今後、どのようなことが情報通信技術で可能になるかを想定し、どのような利用を考えて自身のスキルを身に着けるかを考えるヒントになります。

2.2.1 コンピュータの誕生

電子技術を用いて情報を高速に処理しているのが現代のコンピュータです。このようなことが始まったのは 1940 年前後からで、大量の数値計算を必要とする科学領域などで利用されました。

当初は真空管という白熱電球に似た仕組みを利用したため、大きくて寿命や信頼性も低いことが問題でしたが、トランジスタなど半導体を用いることで寿命の問題が解消され利用が広がりました。1960 年代にはメインフレームと呼ばれる大型のコンピュータが企業での事務処理などで利用されるようになりました。1970 年代には、個人がコンピュータを専有してさまざまな用途に利用するパーソナルコンピュータという利用形態が

立ち上がります。(図 4 参照)

2.2.2 半導体技術の進展

コンピュータの利用が拡大した背景にはコンピュータを構成する複雑な回路を1つの半導体チップ上に集積回路技術とそれを用いて構成したマイクロプロセッサが出現したことです。これによりコンピュータを大量に安価に製造することが可能になりました。集積回路技術はその後、急速に発展し、チップ上に構成するトランジスタの数も個々のトランジスタの動作速度もその性能が何桁も変わる技術革新が今も続いています(図 3 参照)。

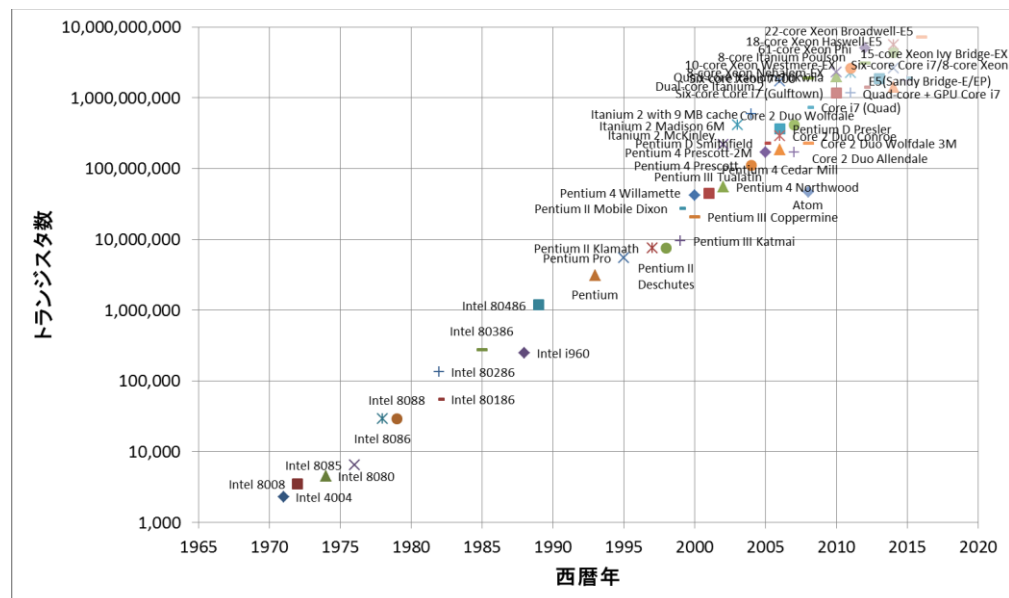


図 3 マイクロプロセッサ上のトランジスタ数

https://en.wikipedia.org/wiki/Transistor_count#Microprocessors

(2017 年 1 月 2 日アクセス)から Intel 社製のプロセッサを抽出してプロット

2.2.3 インターネット

1970 年代からのもう一つの重要な進展はインターネットです。コンピュータどうしをデジタル通信で接続したネットワークを構成して情報の処理と通信が一体化されました。1990 年代にはインターネットが商用にも利用可能になり、パーソナルコンピュータの高度化とも相まって急速に

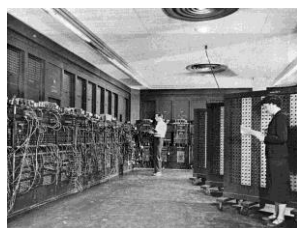
発展普及します。そこでは電子メールや World Wide Web などの情報サービスが展開され、現在では World Wide Web を基盤としてさまざまな情報サービスが提供されるようになりました。さらに、1990 年代から携帯電話網を介して携帯端末がインターネットにつながるようになり、その利用がさらに拡大します。持ち歩くことができ常時インターネットに接続され、使い勝手もよいスマートフォンの出現はパーソナルコンピュータとは異なった情報通信技術の利用を可能にしています。

2.2.4 メディアのデジタル化

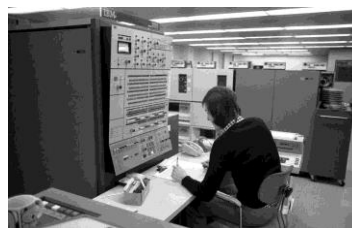
コンピュータやデジタル通信の性能向上に伴い、それまで個別に扱われていた文書、写真、音声、動画などのさまざまなメディアがデジタル化され、一元的に扱えるようになってきたこともコンピュータの利用を拡大しています。

2.2.5 情報技術と社会

他方で、このような世界規模でのコンピュータネットワークの出現はさまざまな社会的問題も引き起こしており、情報通信技術を適正に利用するために情報セキュリティ対策や倫理的な行動が求められています。



* ENIAC 1946



** IBM System/360 1962



*** XEROX Alto 1973



**** Apple II 1977

図 4 コンピュータの発展

* <https://commons.wikimedia.org/wiki/File:Eniac.jpg>

** [https://commons.wikimedia.org/wiki](https://commons.wikimedia.org/wiki/File:Bundesarchiv_B_145_Bild-F038812-0014,_Wolfsburg,_VW_Autowerk.jpg)

/File:Bundesarchiv_B_145_Bild-F038812-0014,_Wolfsburg,_VW_Autowerk.jpg

*** https://commons.wikimedia.org/wiki/File:Xerox_Alto_mit_Rechner.JPG

**** https://commons.wikimedia.org/wiki/File:Apple_II_transparent_800.png (2017/1/3 アクセス)

2.3 パーソナルコンピュータの構成

2.3.1 コンピュータの構成

コンピュータはハードウェア（電子回路などの物理的装置）とソフトウェア（プログラム）で構成されています。

1) コンピュータのハードウェア構成

パーソナルコンピュータは図 5 のような装置で構成されています。

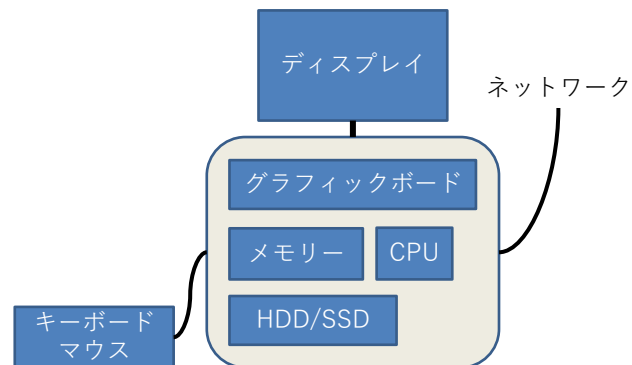


図 5 コンピュータの構成（ハードウェア）

- **CPU（中央処理装置，Central Processing Unit）**：主記憶からプログラムを1ステップずつ読み出し，その命令に応じて，主記憶上のデータや CPU 内のレジスタと呼ばれる記憶装置上のデータに演算を加えたり，転送したり，データに応じて次に実行するプログラム切り替えたりするコンピュータの中核的な装置です。
- **主記憶（メモリ）**：CPU が実行するプログラムや操作するデータを格納する装置です。高速な読み出しや書き込みが求められるので半導体で構成されたメモリが用いられます。
- **外部記憶装置**：大量のプログラムやデータを永続的に保持するための記憶装置です。電源が切れてもデータが保存される磁気ディスク，フラッシュメモリ，CD や DVD ドライブなどの光学的な記憶装置が用いられます。

- **入出力装置**：情報処理のための入力を外部から得たり，結果を提示したりするための装置です。グラフィックボード，ディスプレイ，キーボード，マウス，タッチパネル，プリンタなどが広く用いられています。
- **ネットワーク回線**：インターネットなどコンピュータどうしを接続するためのネットワーク回線です。多くのパソコンでは有線や無線の LAN に接続できるようになっています。

2) コンピュータの動作

コンピュータの基本動作は以下のように極めて単純です。

- 何らかの手段でメモリ上にプログラムや必要なデータを配置します。
- CPU にプログラムの実行開始場所を与えます。
- あとはコンピュータ自身が以下の動作を繰り返します
 1. CPU はメモリ上のプログラムから1ステップ分を読み出し，指示に従って計算，データの転送などを行います。
 - 計算結果をメモリに書き出す場合もあります。
 - 入出力を行う場合もあります。
 2. CPU は実行の対象をプログラムの次のステップに進めます。
 - 実行場所はプログラムによって変化する場合があります。

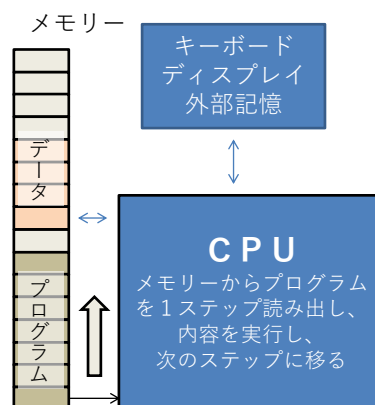


図 6 コンピュータの動作（ハードウェア）

Great Idea！ ストアドプログラム方式

このようなコンピュータの構成は**ストアドプログラム方式**と呼ばれています。複雑な機能の実現をプログラムに委ねることにより，プロ

グラムを変えるだけで多様な利用を実現しています。家電品に組み込まれている小さな制御用のコンピュータからスーパーコンピューターまで共通の構成方法です！

2.3.2 ソフトウェア

ソフトウェアとはコンピュータ上で実行するプログラムのことです。ストアードプログラム方式の採用により、ハードウェアを変えることなく、メモリ上に展開するプログラムを利用したい用途ごとに用意することでコンピュータを様々な用途に使うことができます。その後の技術進歩により、ハードウェアは単純な動作を極めて高速に実現するようになりました。これに対して、用途ごとの複雑な動作はソフトウェアで実現しています。このため高度なソフトウェアではプログラムの行数は数億行にも達しています。

パーソナルコンピュータではソフトウェアは効果的で効率的な運用のためいくつかの形態が取られています。

- **アプリケーションソフト**（例えば Word, Excel）：利用者の具体的なニーズを実現するソフトウェアです。
- **オペレーティングシステム(OS, 例えば Windows)**：ハードウェアとアプリケーションソフトウェアの間に介在して、アプリケーションソフトの実行環境を整えます。

またオペレーティングシステムを補助したり、複数のアプリケーションソフトで共通に利用できるようにしたりする補助的なソフトウェアとして以下のようなものがあります。

- **ミドルウェア**：複数のアプリケーションソフトで共用できる機能（ライブラリ）を実装した補助ソフトウェアです。
- **デバイスドライバ**：プリンタなど機種に応じて装置を駆動する機能をオペレーティングシステムに提供するソフトウェアです。

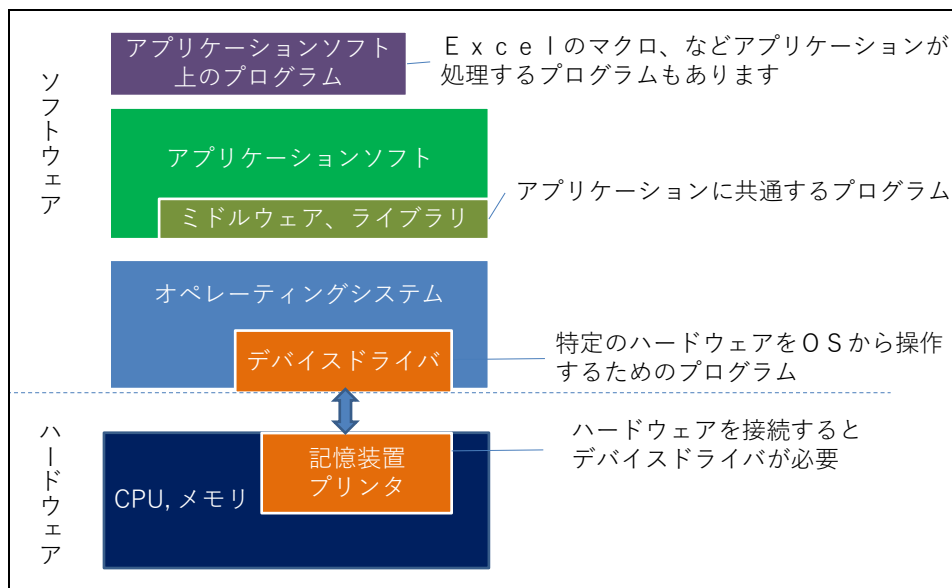


図 7 ソフトウェアの構成

2.4 オペレーティングシステムの役割

2.4.1 OS について学ぶことの意義

オペレーティングシステム（Operating System, OS）について学ぶことにはアプリケーションソフトの操作ほど直接のありがたさはありませんが、

- コンピュータ全般の操作効率が上がります
- コンピュータを適切に管理できるようになります
- コンピュータに生じた問題を把握しやすいです

遠回りのようですが、OS について学ぶことでコンピュータを使った作業の生産性が上がります。

2.4.2 OS の役割

OS はパソコンの動作に関して以下のようなさまざまな役割を担っています。

- アプリケーションソフトの起動、停止、メモリと CPU による実行の割り当て
- ファイルの作成、移動、消去
- アプリケーションソフトのインストール（導入）や削除
- ネットワークでの通信

- キーボードやマウスからの入力
- プリンタへの出力
- ユーザアカウントの作成，ユーザの認証と権限の管理
- ウィンドウの描画，コピーペースト

具体的には図 8 のようなことを担っています。

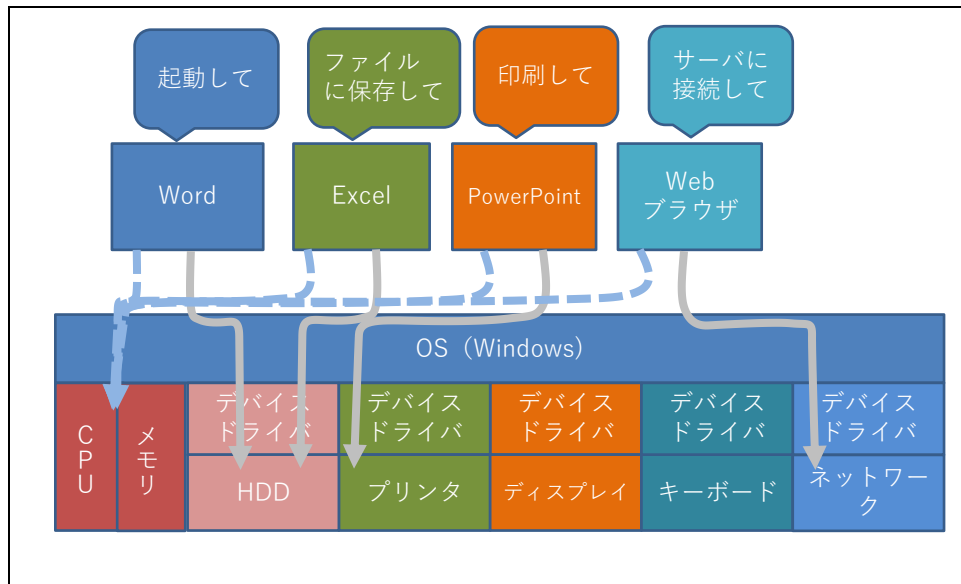


図 8 オペレーティングシステムの役割

Great Idea ! オペレーティングシステム

初期のコンピュータでは、ハードウェアごとに個々にアプリケーションソフトが作られていました。現代ではOSの存在により，開発者にとっては

- ハードウェアもアプリケーションソフトもOSだけを意識して開発すればよくなっています。
- アプリケーションソフト側ではウィンドウの操作などを詳細に記述する必要がなくなります。

ユーザにとっては

- 個々のアプリケーションソフトは同様の操作で使える。

という利点があります。

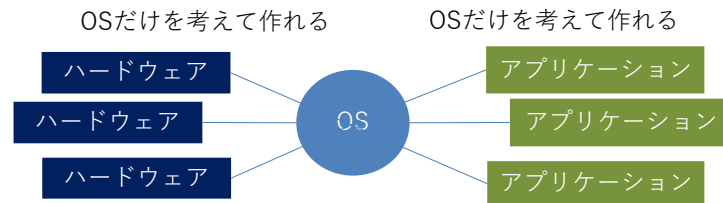


図 9 仲介役としての OS

2.4.3 動作から見たソフトウェアの動き

次にソフトウェアの動きを見て行きましょう。

パーソナルコンピュータを起動すると、

1. まずコンピュータに固定的に設定されているプログラム (BIOS) が起動されます。
2. この BIOS が接続されているディスクドライブなどからオペレーティングシステム(OS)を読み込み、実行を OS に移します。
3. OS が起動し、さまざまな準備が終わるとユーザはウィンドウの操作など OS を介してアプリケーションソフトを起動して利用します。

現代の OS はさまざまな作業を実行しています。パーソナルコンピュータを停止するためには、**シャットダウン**の操作を行ってこれらを適正に終了させた上で電源を切ることが求められます。

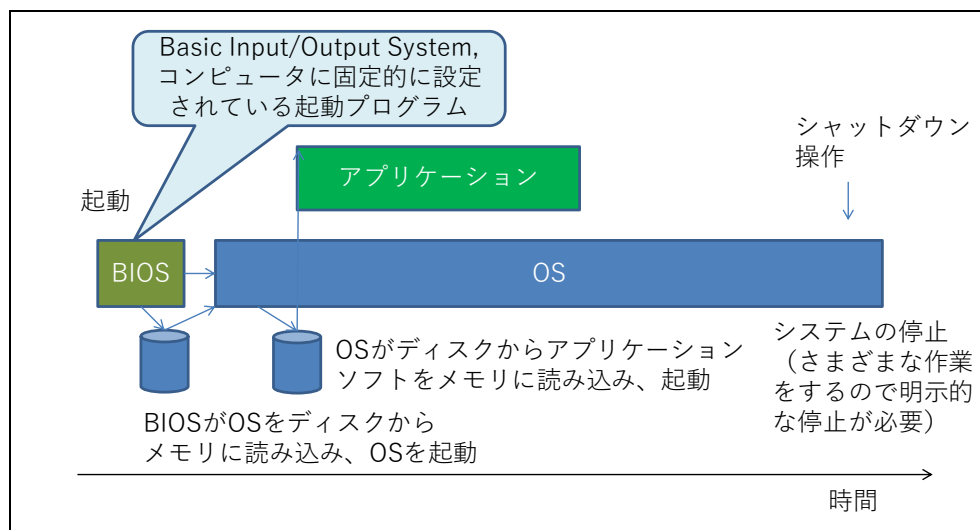


図 10 動作からみたソフトウェアの動き

2.4.4 プログラムの並行実行と CPU とメモリの利用

パーソナルコンピュータではさまざまなアプリケーションプログラムを並行して実行することができます。このために、OS はそれぞれのアプリケーションプログラムと必要なデータ用の領域をメモリ上に配置します。実行にあたっては、それぞれのプログラムの CPU での実行を短い間隔で切り替えて、あたかも複数のプログラムが同時に実行されているようにしています。

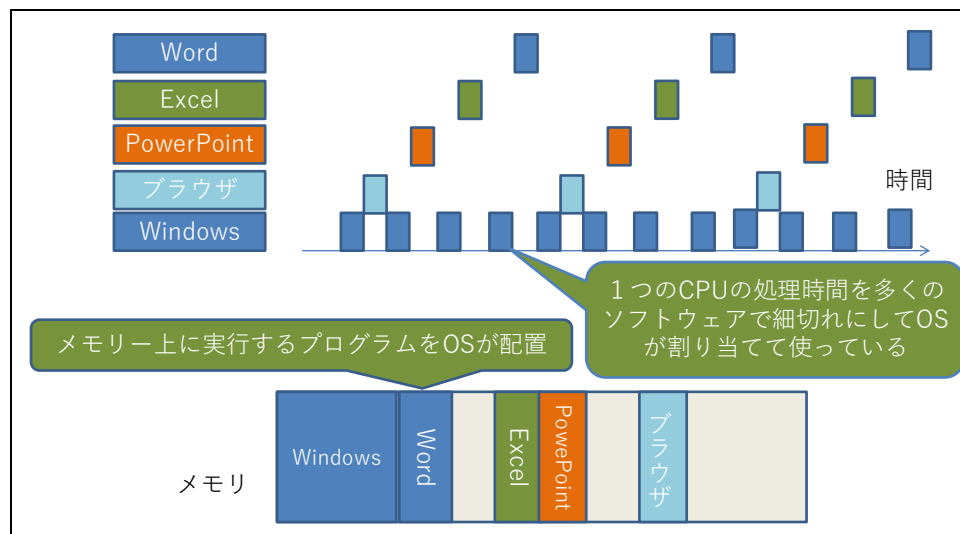


図 11 プログラムの並行実行

2.4.5 OS とユーザインターフェイス

OS は利用者に以下のように統一的な操作環境を提供しています¹。

- ウィンドウの操作
- アプリケーションプログラムの起動, 停止
- メニュー選択
- キーボードの操作
- 印刷
- ファイルの扱い
- コピー & ペースト

ユーザはこれらの操作をアプリケーションごとに覚える必要はなく, 統一的な操作でさまざまなアプリケーションを利用できます。

¹ LINUX での標準的なウィンドウシステムである X-Window は OS とは独立のソフトウェアとして構成されています。

Great Idea ! Window, Icon, Menu & Pointer

WIMP (Window, Icon, Menu & Pointer)はパーソナルコンピュータの標準的なユーザインターフェイスです。1973 年ゼロックスパロアルト研究所 (Xerox Parc) で開発された Xerox Alto で採用されました。図 4 参照。その後, Apple 社の Macintosh に採用され広く使われるようになってパーソナルコンピュータの標準的なユーザインターフェイスとなっています。

2.4.6 OS の利用に習熟する

OS はパーソナルコンピュータを利用する上での基本的な操作を統一的に提供しています。以下のような操作に習熟しましょう。

- ウィンドウの操作,
- プログラムの起動停止,
- ファイル操作,
- 文字入力,
- コピー & ペースト,
- プリンタのプロパティでの設定
- タスクマネージャによるアプリケーションの実行状況の把握や動かなくなってしまったアプリケーションの強制終了

2.5 自立したパソコン利用者に求められるスキル

皆さんはパーソナルコンピュータ（パソコン，PC）を何のために使っていますか？ワードプロセッサ，表計算，プレゼンテーション，写真の整理，ビデオの閲覧，Web の閲覧，メールの読み書きなどさまざまな用途でお使いになっておられるかと思います。これらは基本的には「情報」の「加工」，「蓄積」，「伝達」に使います。

演習 2. パソコンの利用に関して以下の問に答えられますか？

- パソコンを新しい用途に使うためにはどうしますか？
 - アプリケーションソフトウェアの導入し，利用方法を習得するで

しょう。ソフトウェアを変えることによって多様な用途に使えることはコンピュータの最も重要な特質ですが、具体的に何をすればいいのでしょうか。

- パソコンで作成した文書などデータはどこにあるでしょう？
 - パソコン内の HDD や SSD などの記憶装置でしょうか、USB メモリや SD カードなどの取り外しのできる記録メディアでしょうか、あるいはネットワーク上のサーバでしょうか。
 - どこに保存されているのかをどうやって知ることができますか？
 - 保存場所による得失は何でしょうか。
- パソコンでは Windows や MacOS などのオペレーティングシステム (OS) が稼働していますが OS は何をしてくれているのでしょうか？
 - ユーザとして OS についてどのようなことを理解しなければならないのでしょうか。

大学生になって個人でパソコンを専有するようになった方も多いと思いますし、一人暮らしを始めた方も少なくないと思います。このことは**パソコンのユーザとして自立すること**が求められていることを意味します。では自立したパソコンユーザに求められる知識、スキルとはどのようなものなのでしょうか。単にアプリケーションソフトを使えるということだけでなく以下のようなことが求められます。

- 購入、セットアップ、廃棄
 - 予算などの制約と利用の目的に応じてパソコンを選定し、購入できますか。
 - 購入したパソコンをセットアップすることができますか。
 - 不要になったパソコンを適正に廃棄できますか。
- ネットワークへの接続
 - 所有しているパソコンを自宅や大学などで利用できるネットワークに接続できますか。
- 機器の接続
 - プリンタや DVD ドライブなどの機器を接続して利用できますか。
- アプリケーションの導入、更新、削除
 - 必要なアプリケーションを選定し、適正に入手し、導入したり、

更新したり，削除したりできますか？

- セキュリティの確保
 - パソコンを安全に運用するために必要な対策を講じていますか。
 - 利用上の注意事項を知るとともに実施していますか。
 - 問題が生じた場合に適切な対応を取ることができますか。
- データの管理
 - パソコン上で扱うさまざまなデータについて，効果的，効率的に利用するためにどのような方法を取ればいいでしょうか。
 - データを保存するディスクドライブやネットワーク上のサービスを適正に選択できますか。
 - パソコンやディスクの故障に備えてデータの複製（バックアップ）を行っていますか。
 - 重要なデータを適正に保護する手段を知っていますか。
- ユーザの管理
 - パソコンを家族や友人と共用する際に，適正なユーザの権限の設定を行っていますか。
- OS やアプリケーションソフトの利用
 - OS の機能を理解し，効果的，効率的に使用できますか。
 - アプリケーションソフトやネットワークサービスについて効果的，効率的に使用できますか。

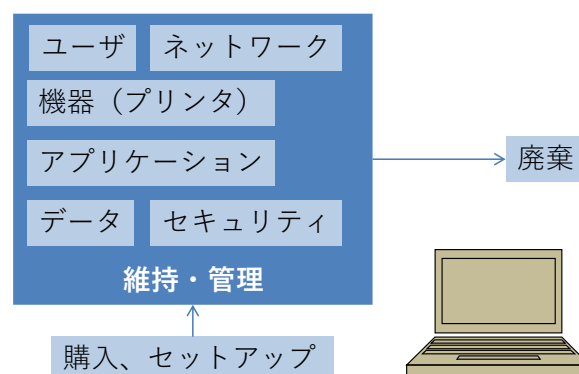


図 12 自立したパソコンユーザの求められるスキル

このような能力を一度に身に着けることは難しいかも知れません。パソコンの構成（ハードウェア，ソフトウェア）やネットワークについての基礎知識を持つことにより系統的に理解できパソコンの運用能力をより効

率よく身に着けることができます。すなわちコンピュータを上手く使うために以下を目指しましょう。

- コンピュータとネットワークの仕組みの基本を知る
- 基本的な操作を再確認し，効率的で効果的な操作方法を学ぶ
- パソコンの用途として大学での学びで要求される利用方法を知る
- 何ができるようになっていけばいいか，自身の理解やスキルを継続的に見直し，より効率的で効果的な利用方法を主体的に学ぶ

2.5.1 コンピュータを選ぶ

パーソナルコンピュータを選ぶときにはどのようなことを考えればいいでしょうか。

- CPU の性能は何に影響しますか？
- メモリの容量は何に影響しますか？
- HDD の容量は何に影響しますか？
- ネットワークや電源への要求は？
- ディスプレイの性能への要求は？
- ノート PC の駆動時間への要求は？

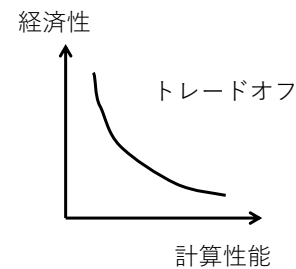


図 13 トレードオフ

あなたがしたいこととの関連はどうでしょうか

- 使うアプリケーション，データ量は？
- 使う場所，機会は？
- 他の機器との連携は？
- 使い方をどうやって学ぶ？
- データ移行は？

ものごとを決定する際にはトレードオフ（あちらを立てれば，こちらが立たずという関係）があります。たとえば費用と性能，可搬性のよさと使う時の操作性の良さなどです。これらを勘案して適切な決定を行うことが求められます。

2.6 キーボードのタイピング

パーソナルコンピュータを扱う上でキーボードのタイピングは人から

コンピュータへの情報伝達の鍵となるプロセスです。スマートフォンの普及により、キーボードよりもタッチパネルの操作の方が多くなっていますが、大量の情報を扱う場合にはキーボードの操作は重要です。タイピングについての理解と、実際に訓練してのスキル向上はパーソナルコンピュータを使う作業の生産性を大きく向上させます。ここではタッチタイピング、英文のタイピング、キーボードを用いた Windows の操作などを紹介します。

2.6.1 タッチタイピング

タッチタイピングとはキーボードを見ることなくタイピングできることです。視線を移動することなくスクリーン上の入力結果だけを見ながら入力できるためタイピング作業を効率的に行えます。



図 14 視線移動の少ないタッチタイピング

タッチタイピングの基礎は標準的な指の使い方です。キーボードの縦の列ごとにそれを打つ指を図 15 のように決めておきます。例えば左端の「1」, 「q」, 「a」 「z」 は左手の小指で打ちます。使わない指は2列目に置くようにします。これをホームポジションと呼びます。人差し指のホームポジションが分かやすくなるようにキー「F」と「J」には小さな突起が付けられています。

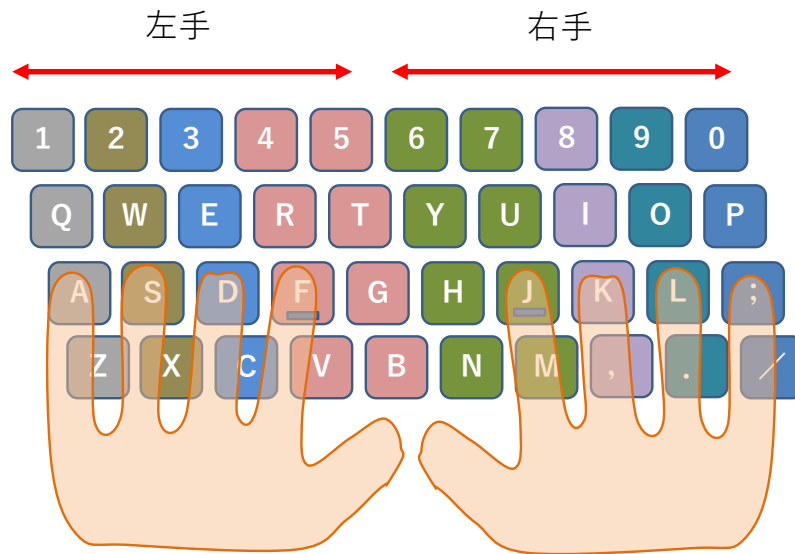


図 15 タッチタイピングのホームポジション

もちろん、タッチタイピングは練習しなければできるようにはなりません。タイピングの練習のための種々のソフトウェアがあります。無料で利用できるものもあるので活用して練習しましょう。情報環境機構の教育用コンピュータシステム端末には **TypeTrainer** というソフトが導入されています。

なお、現在のキーボードは下から 3 段目の文字並びから **QWERTY** 配列と呼ばれたりしています。タイプライタの開発の歴史の中でいろいろな並びが試され現在の配列が主流を占めるようになったそうです。タイプライタの歴史については[2]を参照ください。

2.6.2 英文タイピングの約束

大学では英文をタイピングすることも多くなります。**Word** などのワードプロセッサソフトは英文がいくつかのルールに従ってタイピングされることを前提に文書を整形したり、文法のチェックをしたりします。基本的なルールを身に付けておきましょう。

- **英文は半角文字**でタイピングしてください。
- 英文の**大文字**は
- 図 16 のように **Shift** キーを押しながら文字キーを打つことで入力できます。

- Word で文章を書く場合は、**パラグラフ（段落）を変える時のみ Enter キーで改行**してください。パラグラフ中は行の右端で改行するために Enter キーを押さないでください。
- 文を区切るための**ピリオド、カンマなどの区切り記号の後には空白を1文字**入れてから次の文を始めてください。ピリオド、カンマの手前に空白は入れません。
- 数字の小数点や3桁区切りのカンマは両側に空白を入れません。
- ハイフン (-) の前後には空白を入れません。²
- 引用符 ‘ ’ や括弧 () で文を囲うときは囲われた**内側には記号と文字の間に空白を入れません。記号と外側の文字の間には空白を**入れてください。



図 16 大文字は Shift キーを押しながら

2.6.3 CapsLock と NumLock

英文で大文字ばかりを打ちたいときがあります。このために使うキーが CapsLock キーです。Shift キーを押しながら CapsLock キーを押すと以後、アルファベットキーを押すだけで大文字が入力できるようになります。**知らずにうっかりこの操作をすると突然、大文字ばかりが入力されるようになり困ってしまいます。**もともにもどすには同じ操作、Shift キーを押しながら CapsLock を押します³。

数値データの入力などで数字ばかりを続けて入れたいときに使うのが NumLock です。やはり間違っって押してしまうと困ってしまいますので、動作を確認しておきましょう。

² ハイフン以外に横棒型の文字がいくつかあり、長さが異なります。Word ではハイフンの前後に空白を入れると自動的に少し長いダッシュ (–) になります。

³ キーを押すたびに状態が交互に代わる動作をトグル(toggle)と言います。Word などでは Ins キーは文字を挿入するモードと上書きするモードの切り替えに使います。

2 コンピュータの基礎



図 17 Caps Lock, NumLock に注意, ノート PC では NumLock は別の所に設けられています

2.6.4 Windows のキーボードでの操作

Windows でマウスを用いて行っている操作のいくつかはキーボードでも行えます。キーボードから手を離すことなく操作できるようになるので作業効率が高くなります。代表的なものとして以下の操作をまずは身に着けましょう。以下 Ctrl-~ とは図 18 のように Ctrl キーを押しながら~というキーを押すという操作です。

- コピー (Ctrl-C) / カット (Ctrl-X) / ペースト (Ctrl-V)
- もとに戻す (Undo, Ctrl-Z), やり直す (Redo, Ctrl-Y)
- Tab キーによる入力項目の移動
- タスクマネージャを呼び出すための Ctrl + Alt + Delete, Ctrl キーと Alt キーを押しながら Delete キーを押すことでタスクマネージャの呼び出しなどができます。



図 18 Ctrl キーの利用, Tab キーの利用

2.6.5 その場議事録

速いタイピングが役に立つ場面として会議での議事録作成があります。

会議の場で会議の内容をその場でタイプし、会議をしながら議事録を作成して行くのです。その際、P Cの画面をプロジェクタで投影し、議事録の内容を確認しながら会議を進めます。会議の終了時には、何が決定され、どの議論が積み残され、誰がどういう用務を担当するのかなどをP Cの議事録を見ながら確認していきます。このようなやり方はその場で議事録を作成するので「その場議事録」とでも呼ぶべき技法です。

ラーニングコモンズを使ったグループワークなどでぜひ、トライしてみてください。

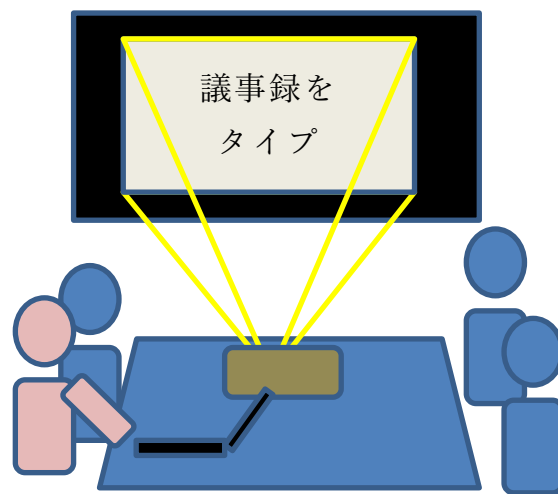


図 19 会議の場で議事録をとることで、情報共有。
タイピングは必須のスキル。

Great Idea! かな漢字変換

民族学者の梅棹忠夫が 1969 年に書いた「知的生産の技術」[1]は今でも読まれている名著ですが、その中で日本語をローマ字書きすることや、カナタイプライタを利用することなど、当時の技術でいかに日本語を扱うかを論じています。アルファベットと数字、若干の記号だけという文字種の少ない英文が早くからタイプライタを用いて文書を作成することでオフィスでの仕事を支援できていたことに対して、漢字かな混じり文を用いる日本語を機械で扱うことには困難がありました。

これが転機を迎えたのは 1978 年に森健一氏らによって開発された「かな漢字変換」という技術を用いた東芝の日本語ワードプロセッサ JW-10 の発表です[3]。そこでは今はもう当たり前の技術になっています

が、かなで入力し、コンピュータの力を借りて漢字に変換することが行われ、同音異義語の多い日本語に対して変換候補を使用頻度などで決めて提示するというものでした。

もし、この技術の開発が遅れていたら私たちはローマ字やカナ文字で日本語を書くようになっていたかもしれません。そして、それは私たちの文化へのはかりしれない影響を与えていたかもしれないのです。現在ではさまざまな言語の文字でコンピュータを利用できます。日本語ワードプロセッサの開発がその先鞭をつけたという意味で人類への文化的貢献も大きいのではないのでしょうか。

2.7 ソフトウェアのライセンスと保守

2.7.1 ソフトウェアの保護

先に見たようにコンピュータはハードウェアとソフトウェアで構成されています。現代のソフトウェアは大きなものならプログラム（ソースコード）の行数が数億行にもものぼります。当然、このようなソフトウェアを開発し、保守するためには多くの人々がかかわり、多大なコストを要しています。

一方、ソフトウェアは単なるデータですので容易に複製できてしまいます。ソフトウェアの作成者の権利を守るために、**ソフトウェアは著作物として作成者の権利を著作権法により保護**しています。

ソフトウェアの利用にあたっては、利用者は著作権者から**利用許諾（ライセンスング）**を受ける形で契約を結びます。したがって、ソフトウェアの利用者はこの契約を遵守することが求められます。

コンピュータにソフトウェアをインストールする際には、利用許諾の要件が示され、これに同意した上でソフトウェアがインストールされることが多いはずです。利用許諾の内容として、利用者に何が許されているのか、提供者がどのような権利を持つのか、利用して生じた問題などに対して、提供者が責任を負わないこと（免責事項）などが示され、これに合意して利用を契約することになります。

2.7.2 ライセンスの形態

ソフトウェアのライセンスにはさまざまな形態があります。

1) コンピュータの運用形態によるもの

- **導入するコンピュータごとのライセンス**：皆さんがお持ちのパーソナルコンピュータにソフトウェアをインストールする場合に一般的に見られるライセンスの形態です。
- **フローティングライセンス**：大学に設置している端末のように多数のコンピュータを多くのユーザが使う場合、どの端末でもソフトウェアを使いたい、すべての端末で同時に稼働させることはない場合があります。このような場合に行われる契約として、どの端末で稼働させてもよいが、同時に実行できるコンピュータの台数を制限するライセンスです。ライセンスを管理するサーバを置いて、そのサーバから許諾を得た端末でだけ実際にソフトウェアが起動します。
- **サイトライセンス**：大学や学部などがその組織内で何台のコンピュータにインストールしてもよいという形態のライセンスです。

2) コンピュータの利用目的によるもの

- **学術機関向けのライセンス**：大学など、教育や研究といった公共性のある活動での利用に配慮して商業的な利用は行わないなど利用目的を限定して、安価でソフトウェアが提供されるライセンスです。

3) 無償で利用できるもの

- **フリーソフトのライセンス**：無償で利用できるソフトウェアも数多くありますが、これも多くの場合は利用許諾を受ける形で利用することになります。利用許諾に当たっては利用条件が課せられていることがあります。

2.7.3 ソフトウェアの保守の必要性

コンピュータにソフトウェアを導入した場合、いくつかの理由でこれを継続的に保守する必要があります。利用者としては提供者からの更新情報にしたがってソフトウェアの更新を行う必要があります。

- ソフトウェアが抱える脆弱性：ソフトウェアは外部からの攻撃に対する脆弱性の問題を抱えています。特に多くの人が頻繁に利用するソフトウェア（オペレーティングシステム、Web ブラウザ、オフィスソフトなど）は攻撃の対象ともなりやすいとされています。
- 他のソフトウェアへの依存：アプリケーションプログラムはオペレーティングシステムや共通のライブラリなどに依存して稼働します。オペレーティングシステムやライブラリが更新された場合、アプリケーションプログラムが新しい状況では適正に稼働しないことも生じます。

また、すでに提供者の側で保守されていないソフトウェアを継続的に利用することは脆弱性を放置することになるので危険です。

2.7.4 ソフトウェアによる情報収集

最近のソフトウェアは提供者のサーバと連携して動かすものが多く、サーバへの連携などに際してソフトウェアの利用者の情報を提供者が収集するものも少なくありません。利用許諾の際に、どのような情報を、誰が、どのような目的のために集めるのかが示されます。ソフトウェアの導入に際してはこれらの点にも留意する必要があります。

演習 3. パーソナルコンピュータのハードウェア

あなたがパーソナルコンピュータを保有しているならご自身のコンピュータについて以下の事項を調査してください。

- 搭載されている CPU
- 主メモリの容量
- 外部記憶装置（HDD や SSD）の種類や容量
- ディスプレイの解像度
- 外部に機器を接続するためのインターフェイス

演習 4. パーソナルコンピュータのソフトウェア

あなたがパーソナルコンピュータを保有しているならご自身のコンピ

ュータについて以下の事項を調査してください。

- オペレーティングシステムの種類，バージョン
- オフィスソフトの種類，バージョン
- ウイルス対策ソフトの種類，バージョン
- 上記のソフトウェアの利用許諾条件
- 上記のソフトウェアの更新方法

演習 5. オペレーティングシステムの役割

オペレーティングシステムの役割について簡単にまとめてください。

演習 6. プリンタの利用

あなたがパーソナルコンピュータを保有していてプリンタを接続しているなら以下の事項を調査してください。

- プリンタの接続方法
- プリンタの設定の調整方法

演習 7. 記録メディアについて

CD, DVD, USB メモリ, SD カードなどについて以下の点について調査してください。

- メディアの種類と容量，書き込みが行えるかどうか
- メディアを介したウイルスへの感染のリスクと対策
- メディアによる情報漏洩のリスクと対策

演習 8. タイピング

キーボードのタイピングについて，以下を確認してください。

- 大文字など複数のキーを用いて行う入力の確認
- タッチタイピングの指使い
- 英文タイピングのルール
- キーボードを用いたコピー＆ペーストのしかた

参考文献

- [1] 梅棹 忠夫: 知的生産の技術 (岩波新書) 岩波書店 (1969)
- [2] 安岡孝一, 安岡素子著: キーボード配列 QWERTY (クワーティ) の謎, NTT 出版, 2008.
- [3] 情報処理学会, コンピュータ博物館, JW-10,
<http://museum.ipsj.or.jp/computer/word/0049.html> (2015 年 5 月 11 日アクセス)

3. ネットワークの基礎

3.1 学習の目的

インターネットで世界中のコンピュータが接続されることは大きな社会変革をもたらしています。さまざまなサービスがインターネットを通じて提供されるようになりましたが、同時に多くの問題も生じており、情報セキュリティや情報倫理について正しい知識を身に付け、それに従って行動することが求められています。本章では、このような状況を理解し、実践するために

- LAN やインターネットなどネットワークの基礎的な知識
- ICT の利用に際してのリスクと対策

について学びます。また、現代ではさまざまなメディアを用いてコミュニケーションが行われますが、

- フォーマルなコミュニケーションで用いられる電子メールについて

京都大学でのサービスと学生の皆さんに求めたい事項を説明します。

3.2 仮想社会の安全運転

インターネットにより世界中のコンピュータや人々がつながった巨大な仮想社会が形成されています。コンピュータは知的能力を飛躍的に向上させる道具、言うなれば知的パワードスーツ、あるいは交通で例えるならジェット機や大型トラックといったところでしょうか。そしてインターネットは、知的パワードスーツに身を固めた人々が自在に走り回る仮想社会のハイウェイです。



ですので、様々な凄ことができるようになっていますが、仮想社会で安全運転しないと大変なことになるってしまいます。仮想社会では実社会以上に犯罪など悪意のある行為も横行していますし、多数の人々によるバッ

シング行為など実社会では生じにくい問題も深刻です。

このような仮想社会で活動するために情報ネットワークやそこで提供されているサービスについて基本的な知識を持つことが求められます。

3.3 情報セキュリティ e-Learning の受講

情報倫理や情報セキュリティについては数多くの学ぶべき事項があり、この授業でも折に触れて必要な注意喚起はしますが、系統的に知識を身に着けて頂くために情報環境機構が提供する情報セキュリティ e-Learning での学習が求められています。これを完了して頂くことも本章での学習の必要事項となります。

情報セキュリティ e-Learning を修了してください

3.4 ローカルエリアネットワーク (LAN) の基礎

パソコンなどのネットワークへの接続は

- 有線 LAN, 無線 LAN (Wi-Fi) などへの接続と
- インターネットへの接続

の2段階になっています。以下、LAN とインターネットの基礎的事項について説明します。

3.4.1 LAN の種類

1) 有線 LAN

イーサネット (Ethernet) と呼ばれます。

- 同じ RJ45 型コネクタを使いますが、表 1 に示すように通信速度の異なるいくつかの規格があります。通常、自動的にどの規格に従うか設定されます。



図 20 RJ45 型コネクタ

3 ネットワークの基礎

- 規格によって要求されるケーブルの品質（カテゴリと記載されます）が異なっています。
- LAN の機器ごとにあらかじめメーカーによって固有の番号(MAC アドレス) が定められています。

表 1 主な有線 LAN

規格	通信速度	使用するケーブル
10 BASE-T	10 Mbps	カテゴリ 3
100 BASE-TX	100 Mbps	カテゴリ 5
1000 BASE-T	1000 Mbps	カテゴリ 5 e

2) 無線 LAN (Wi-Fi)

無線 LAN は使用する電波の周波数帯, 通信速度によっていくつかの規格があります。通常, パソコンとアクセスポイントの間で自動的に適用する規格が決定されます。

表 2 主な無線 LAN(Wi-Fi)の規格

規格	周波数帯	最大通信速度
802.11a	5GHz 帯	54Mbps
802.11b	2.4GHz 帯	11Mbps
802.11g	2.4GHz 帯	54Mbps
802.11n	2.4GHz 帯, 5GHz 帯	600Mbps
802.11ac	5GHz 帯	6.9Gbps

SSID, ESSID: 同じ空間で複数の無線 LAN が混在できます。SSID, ESSID は個々の無線 LAN グループを識別するために付けられている名前です。

3) 無線 LAN での暗号化

無線 LAN では電波が空間中を伝わるため容易に他者に盗聴されます。これを防ぐために暗号化が行われます。暗号化された無線 LAN には SSID に加え暗号化キーが接続に必要です。自宅などでアクセスポイントを設置する場合は WEP など脆弱性が指摘されている暗号化方式は使わないでください。

表 3 無線 LAN の暗号化の規格

暗号化の種類	備考
WEP	初期の方式，脆弱性が指摘されている
WPA	WEP を改良，TKIP と呼ばれる暗号化方式を利用
WPA2	WPA の強化版，より強力な AES 暗号に対応

暗号化を行わない LAN を安全に利用するためには VPN (Virtual Private Network) 接続などを用います。

4) 京都大学での無線 LAN アクセスポイントへの接続

大学の Wi-Fi アクセスポイントでは幾つかの方式が併用されています。

- KUINS Air：ECS-ID, SPS-ID で接続します。学内限定の Web サイトなども閲覧できます。
- EDUROAM：大学間で無線 LAN を相互に利用できるサービスです。EDUROAM 専用のアカウントを KUINS から取得して使ってください。EDUROAM に参加している他大学でも使えます。
- 公衆無線 LAN：NTT ドコモ，KDDI(au)，SoftBank が提供している無線 LAN サービスを学内のアクセスポイントで使えます。

5) VPN (Virtual Private Network)

インターネット上で、仮想のプライベートなネットワークを構成する方法です。京大では PPTP 方式などの VPN を提供しています。

- 学内オープン設定の KUINS III 情報コンセントからの接続

3 ネットワークの基礎

- 学外からの学内ネットワークへの接続に使います。

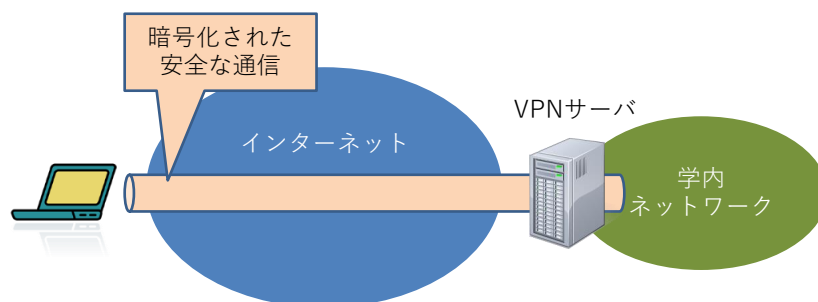


図 21 VPN の仕組み

3.5 インターネットの基礎

インターネットを介して図 22 のように様々な機器が接続されています。以下ではインターネットについて基礎的な事項を学びます。

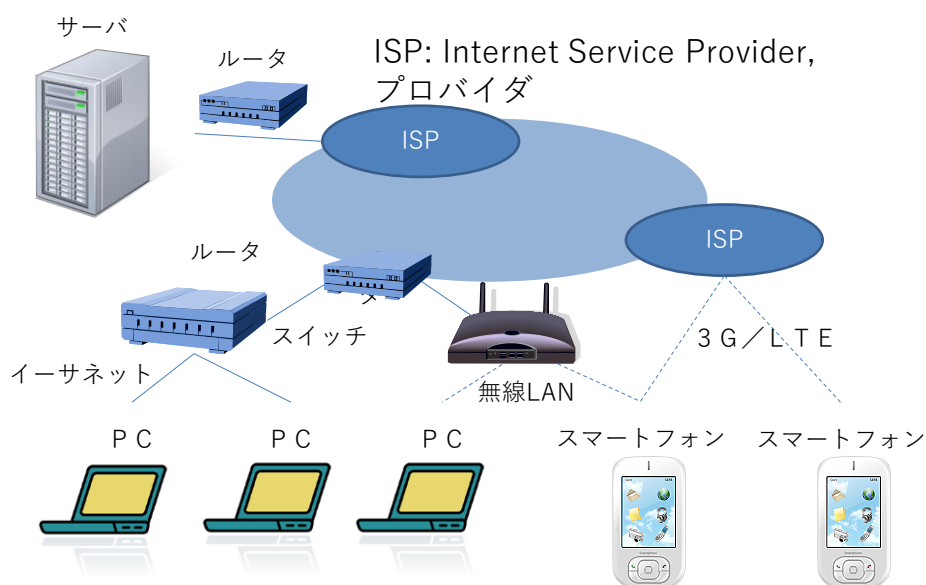


図 22 ネットワークを構成するさまざまな機器

3.5.1 インターネットの仕組み 1: TCP/IP

インターネットはLAN などさまざまな種類のネットワークを介して使える共通の通信の仕組みを提供しています。情報はパケットという単位で

3 ネットワークの基礎

配送されます。これにより 1 つの回線で多くの相手と並行して通信できます（図 23 参照）。

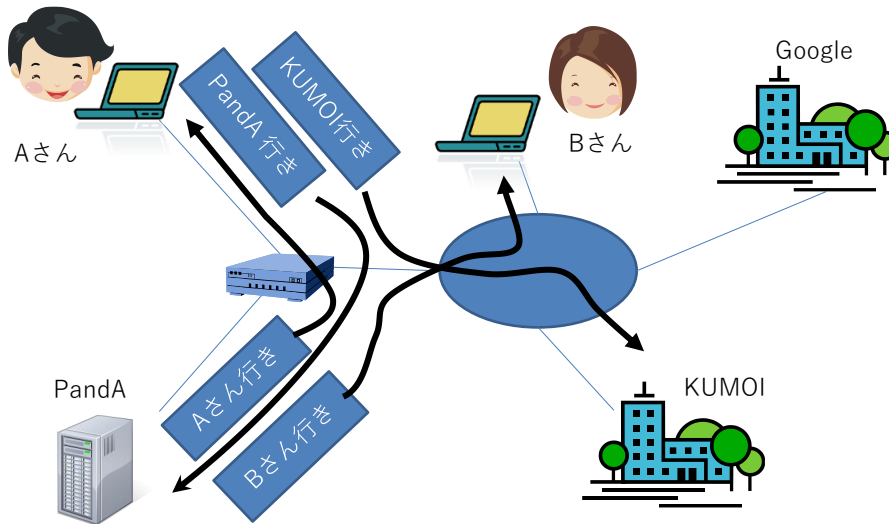


図 23 パケット通信

インターネットは末端から末端へのパケットの配送が基本です。

- それぞれの末端の機器を識別するためのアドレス（番地）が「IP アドレス」です。
- IP (Internet Protocol)：パケットレベルの配送だけを担当します。最善の努力で配送しますが、パケットの到着や順序などは保証しません。
- TCP (Transmission Control Protocol)：末端から末端へデータが順序を守って確実に届くことを実現しています。IP レベルではこのことは保証されませんので、再送や順序の復元などを行っています。同じコンピュータで複数の TCP での通信を行うために IP アドレスに加え、ポート番号で識別します。
- HTTP (Hypertext Transfer Protocol)：TCP を用いて World Wide Web のサービスを行うためのプロトコルです。

このほかにも、TCP/IP を基盤にさまざまなプロトコルが定められています。

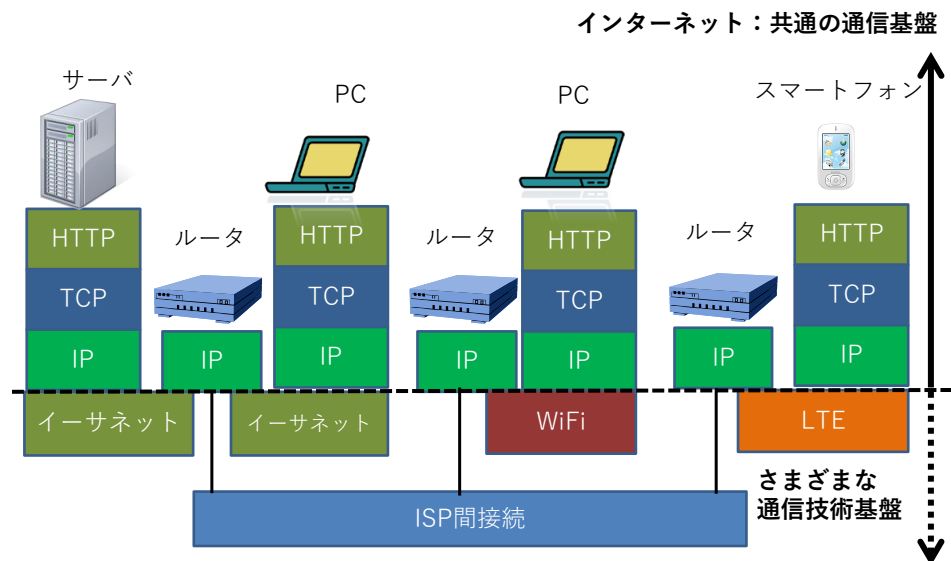


図 24 インターネットを構成する通信規約（プロトコル）

Great Idea! インターネットプロトコル

インターネットの基盤であるパケットを配送するインターネットプロトコル（IP）は

「最善の努力で配送」する。到着すること、パケットの順番を守ることとも保証しない。

と定められています。

ネットワークの途中では、このことだけを実現すればよいので、中継の仕組みが簡単になります。長いメッセージを、複数のパケットを使って、到着や順番を保証して通信することは末端での役目とされ Transmission Control Protocol (TCP)が担います。IP と TCP がインターネットを代表する通信規約であるので TCP/IP と称されます。

このような仕掛けで利用者がさまざまなサービスのためのプログラム（ソフトウェア）を開発できることがインターネットの発展を支えました。

3.5.2 グローバル IP アドレスとローカル IP アドレス

IP アドレスには全世界で系統的に割り当てて、インターネット上でパケットが配送されるグローバルアドレスのほかに、閉じた範囲で自由に使えるかわりに、その範囲を超えて配送してはいけないローカルアドレスが定められています。

京都大学のネットワーク KUINS は以下の 2 系統で運用されています。

- **KUINS II:** Web サーバなどを設置するための LAN でグローバル IP アドレスが与えられます。
- **KUINS III:** パーソナルコンピュータなどを接続するための LAN でローカル IP アドレスが与えられます。

前者は外部にサービスすることが可能ですが、その分、攻撃なども受けやすいという危険性があります。後者は外部から直接はパーソナルコンピュータにアクセスできないため、より安全に運用できます。KUINS III に接続したパーソナルコンピュータから外部の Web サーバなどにアクセスできるのは、そのための中継の仕組みが KUINS II と KUINS III の間に設けられているからです。

一般家庭(個人)で使う有線、無線のルータでも、それに接続する PC やスマートフォンには基本的にローカル IP アドレスが使われています。

3.5.3 IP アドレスの割り当て

IP アドレスは個々のコンピュータを識別し、パケットを配送するためのアドレスですからネットワーク内で（グローバルアドレスなら世界的に！）系統的に割り当てることが求められます。皆さん方が使うパーソナルコンピュータでは手作業で明示的に IP アドレスを割り当てるとはほとんどないはずです。これはコンピュータを起動したときに IP アドレスを自動的に割り当ててくれるサーバと通信してアドレスを取得するようにネットワークが運用されているためです。このためのプロトコルを DHCP と呼びます。

3.5.4 インターネットの仕組み 2:ドメインネームシステム (DNS)

IP アドレスは分かりにくいので、人に分かりやすい命名としてドメイン名を使います。ドメイン名から IP アドレスを検索して決定する分散的なデータベースがドメインネームシステム DNS です。ドメイン名から IP アドレスを検索してくれるサーバをネームサーバ (DNS サーバ) と呼びます。接続するネットワークによってネームサーバを自動的に取得してくれる場合と、明示的に設定しなければならない場合があります。

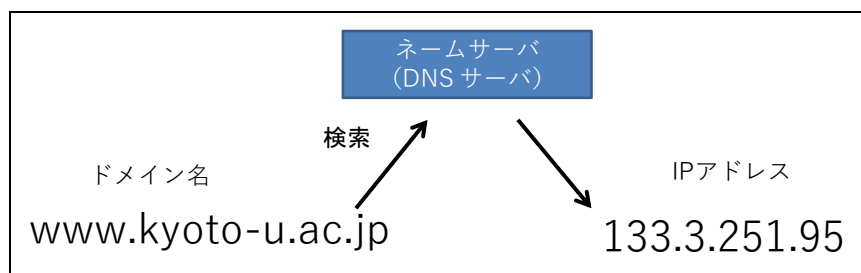


図 25 DNS によるドメイン名から IP アドレスへの変換

3.5.5 WWW の仕組み

World Wide Web は http というプロトコルを用いて Web ページにアクセスする仕組みです。その特徴は

- HTML という形式でページが記述されています。
- URL という記法でウェブサイトやその上に置かれたページなどを統一的に記述します。
- ハイパーリンクとしてページ内で他のページへのリンクを書くことができます。

Web サイト (サーバ) にアクセスするためのアプリケーションソフトは「Web ブラウザ」と呼ばれます。主な Web ブラウザとしては

- Microsoft 社が提供する Internet Explorer や Microsoft Edge,
 - Mozilla Foundation が提供する Mozilla Firefox,
 - Google 社が提供する Google Chrome,
 - Apple 社が提供する Safari
- などがあります。

3 ネットワークの基礎

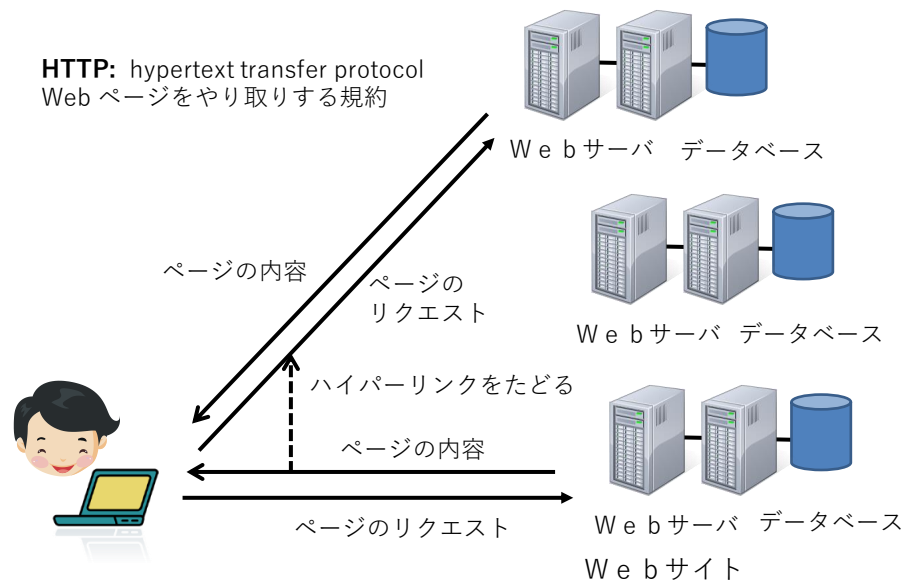


図 26 World Wide Web の仕組み（ハイパーリンク）

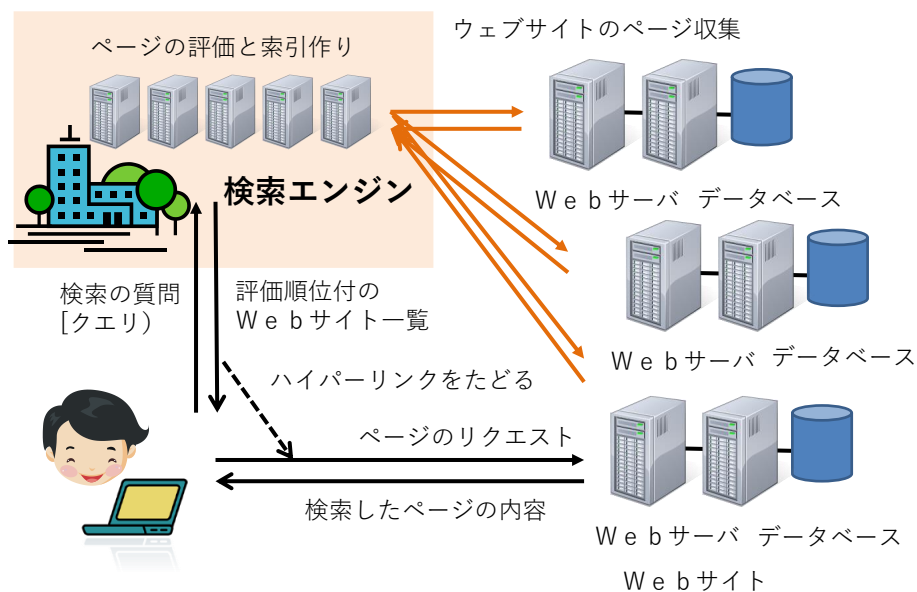


図 27 World Wide Web の仕組み（検索エンジン）

現在では World Wide Web 上で様々なサービスが提供されています。例えば google などの検索エンジンでは世界中の WWW サイトから情報を自動的に収集し、そのページの重要性を評価したり索引を作ったりして、検索の要求に対して該当するページの候補を提供しています。

3.5.6 電子メールの仕組み

電子メールはインターネット上の電子メールサーバを相互に接続してメッセージを配送する仕組みです。ユーザがメールを送受信する方法としては

- 電子メールサービスを提供するサイトの Web ページを介して Web ブラウザでアクセスする
- Microsoft 社の Outlook や Mozilla Foundation の Thunderbird など パーソナルコンピュータでメールソフトを起動してアクセスするという 2 種類の方法が主に使われます。

後者では

- メールを受信するためのプロトコルとして
 - POP (Post Office Protocol),
 - IMAP (Internet Message Access Protocol)の 2 通りがあります。
- 送信するためのプロトコルとしては SMTP (Simple Mail Transfer Protocol) が用いられます。

メールソフトを用いるには、これらについてメールサービスの指定するサーバとプロトコルを設定しなければなりません。

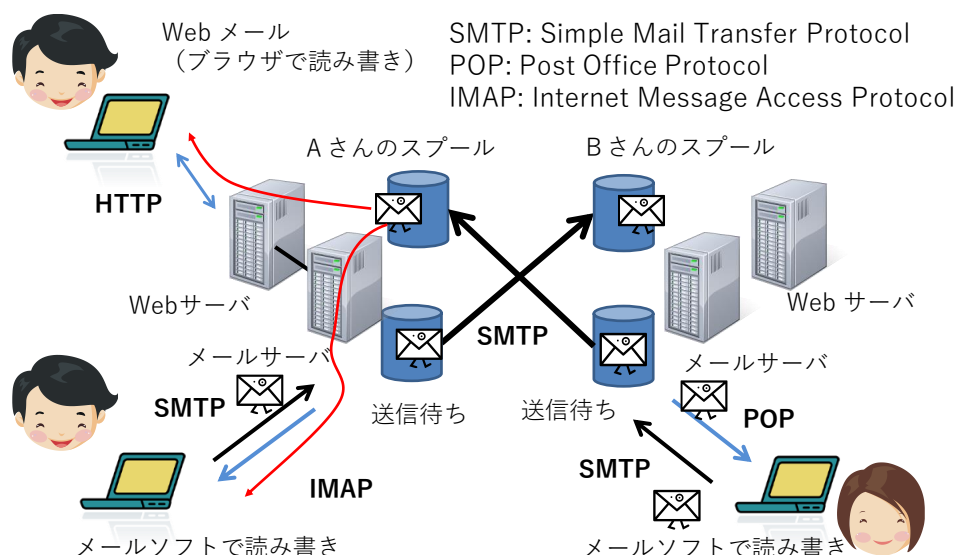


図 28 電子メールの仕組み

3.5.7 インターネットでの暗号の利用

TCP/IP によるインターネットの通信は平文で行われます。通信経路上でメッセージは特に保護されていません。内容を途中で覗き見されないためには暗号を利用します。

- インターネット（WWW）上での暗号通信として SSL（TLS）がよく使われます。
- SSL を使うウェブの URL は“HTTPS”と表記されます。
- POP3S, IMAP4S, SMTPS: それぞれ POP3, IMAP4, SMTP でのメールの通信を SSL を使って暗号化する方式です。POP3 や IMAP4 の 3 や 4 はそれぞれのプロトコルのバージョンを表します。

1) 公開鍵暗号

公開鍵暗号は安全なインターネットの利用を支える極めて重要な技術です。金庫に例えると、閉める鍵と開ける鍵が異なった仕掛けで、しかも片方の鍵や暗号化されたメッセージから他方の鍵を推測することが極めて難しい（途方もない計算量を必要とする）ように作られています。

2つの鍵、A、Bを用い得るとして片方を公開し、他方を秘密にします。例えば

- 鍵A：公開
- 鍵B：秘密

これを用いると

- 暗号通信
 - 公開の鍵Aで暗号化（誰でもできる）→ 鍵Bで復号化（秘密鍵の所有者のみ可能）
- 電子署名
 - 秘密鍵Bで暗号化（所有者のみ可能）→ 公開鍵Aで復号化（誰でもできる）

という運用により、不特定多数の通信者を相手に暗号通信や電子署名を行うことができるのです。

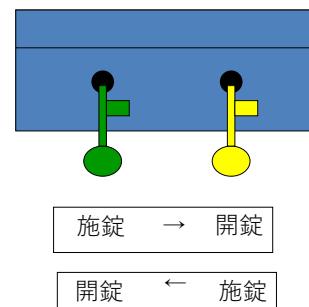


図 29 公開鍵暗号

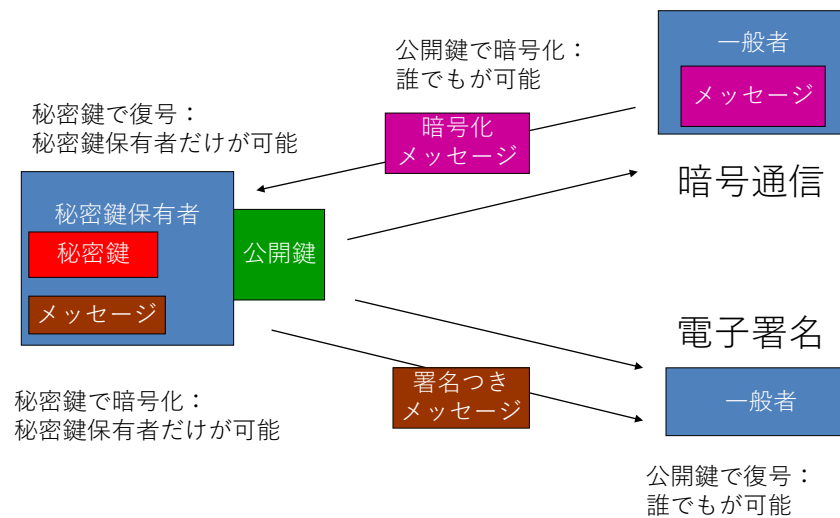


図 30 公開鍵暗号を用いた暗号通信と電子署名

Great Idea! 公開鍵暗号

公開鍵暗号が開発されるまでの暗号方式は暗号化と復号化に共通の鍵を用いていました。このため鍵を受け渡すことが難しく、不特定多数を対象にした暗号通信は困難でした。現在、広く用いられている公開鍵暗号は巨大な素数の積を因数分解することが極めて難しいという原理を用いています。

2) 認証局と証明書

公開鍵暗号を用いて暗号通信ができて、通信する相手が信用できるかどうかは別の問題です。公開鍵の発行者の正当性を知るための仕掛けが「認証局と証明書」で、印鑑証明の仕掛けを電子的に実現したものです。

- 証明書を発行する認証局を作り、その公開鍵を予め何らかの方法で利用者に配布します
- 証明を受けようと思うものは、公開鍵を認証局に登録します。これに対して、認証局は電子署名した証明書を発行します
- 証明を受けたものは公開鍵とともに証明書を利用者に示します
- 利用者は公開鍵の正当性を認証局の署名により確認します

3 ネットワークの基礎

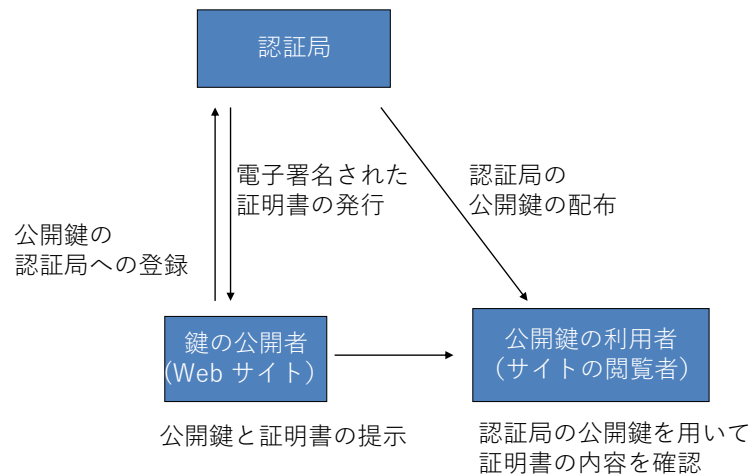


図 31 認証局と証明書

3) SSL 利用と公開鍵証明書の確認

Web サイトへのアクセスを暗号化するのに用いられている仕組みが SSL です。アクセスしているサイトへの通信が暗号化されていることと、そのサイトの公開鍵が適切に証明されていることを確認しましょう。



図 32 WWW での暗号通信と証明書の確認

3.6 パーソナルコンピュータのネットワーク接続

先にも述べたようにパーソナルコンピュータのネットワーク接続は

- LAN への接続

- インターネットへの接続
の2段階になっています。

- まず、有線 LAN、無線 LAN など LAN への接続が行われます。
- LAN 上にあるサーバから、インターネット接続のためのアドレス (IP アドレス) が割り当てられ、その他の必要な情報が提供されインターネットへの接続が行われます。

通常はこの両方は一括して行われますが、LAN 上のサーバと適切に通信ができていない場合や間違った LAN に接続した場合は LAN には接続できているが、インターネットへの接続は行えない、というような状況になります。

3.7 電子メールの利用

3.7.1 京都大学のメールサービス

京都大学のメールサービスとして学生用のメールシステム KUMOI のアカウントが与えられます。また重要な情報が KULASIS や KUMOI に送られます。

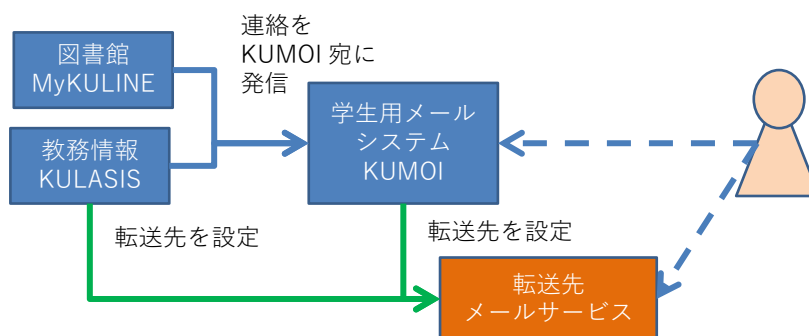


図 33 京都大学のメールサービス

3.7.2 メール利用に際しての注意点

1) メールの確認

大学のメールサービスは大学から皆さんへの重要な連絡手段です。定期

的に（1日に一度程度は）メールの有無をチェックするようにしてください。夏休みなど休業期間中でも連絡を行うことは少なくありません。

2) 安全な運用

メールはウイルス感染や詐欺などにも利用されます。不審なメールの添付ファイルを開くことや、そこに示された Web サイトへのアクセスは行わないようにしてください。

3) 適切なメッセージの構成

教職員はメールによる皆さんからの連絡には様々な問題を感じています。これまでのメール利用（やその他のコミュニケーション手段）では

- 主に家族や友人との私信が中心に
 - スマートフォンなどを使って短いメッセージを発信する
- という使い方をされていたと思います。

これに対して大学でのメール利用は

- 大学での学習を含めさまざまな社会的活動で用いるフォーマルなコミュニケーションです
 - メールアドレスだけで個人を特定することは困難です
 - 連絡する相手（教員など）は数多くの業務を抱えています
- このような状況で確実にメッセージが相手に届くことを心がけてください。具体的には

- 件名を明記してください。
- 受取人を明記してください。（受取人は良く確かめて）
- 差出人を明記してください。
- 用件を受取人が理解できるように書いてください。

演習 9. メッセージについて

あなたがもし以下のピンク色のようなメールを受け取ったらどのような問題を感じるでしょうか？水色のメッセージだと何が良いでしょうか。

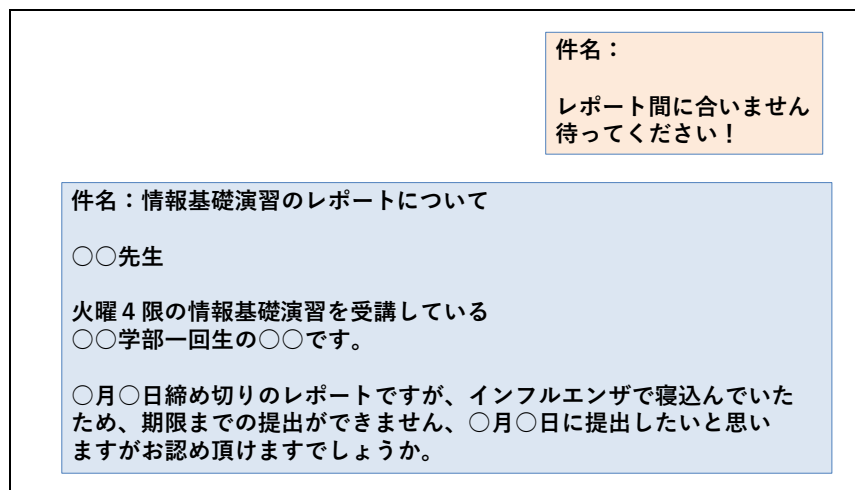


図 34 困ったメールのメッセージ

3.8 情報通信技術 (ICT) の利用とリスク

1) 情報の持つ特性

情報には**様々な人にとって商業的価値**や**プライバシー**など様々な価値があります。一方、デジタル化されネットワーク上で扱われる情報には以下のような特性があります

- 複製が極めて容易です
- 一度、広がると消すことは困難です
- 一度消えると戻りません
- それだけで真偽を判断しにくいです

2) ICT利用が孕むリスクが問題になる理由

インターネットで世界中のコンピュータが繋がり、それを様々な人々が利用していることから ICT の利用はさまざまなリスクを孕んでいますが、その理由は

- ネットワークが世界規模で広がり、高速な通信と情報処理が行われており、
 - 大量の情報の保持されている、そして
 - 様々な意図や価値観を持つ多数の人々が利用している
- ため大きな影響を持つ可能性があるからです。

リスクを無くすことが難しい理由は以下のようなことがあるからです

- コンピュータのソフトウェアは複雑で脆弱性を完全には無くせません
- 利用する人は不注意などで適正でない行動をとってしまいます
- 悪意を持つ人を含め、さまざまな人がインターネットを使っています

3) 情報の何を守るべきか

情報セキュリティの管理では情報について以下の3つの視点で考えています。

- **完全性**：情報が不完全になってしまうリスク
- **可用性**：情報が利用できなくなってしまうリスク
- **機密性**：情報の機密性が失われてしまうリスク

4) リスクの要因

情報に関するリスクを分けると物理的なリスクと人の行動に伴うリスクがあります

- 物理的なリスク：機器の故障、停電などによる停止
- 人の行動に伴うリスク
 - 判断や操作の間違い、サービスの不十分な理解
 - 悪意のある行為
 - 数多くの人々の反応

5) リスクの回避

上記のような情報の持つ特性とリスクについて理解し、情報の取扱いと情報機器の操作について適正な行動をとる必要があります。詳しくは情報セキュリティ e-Learning で学習してください。

演習 10. パーソナルコンピュータのネットワークインターフェイス

あなたがパーソナルコンピュータをお持ちなら以下について調査してください

- 有線 LAN のインターフェイスと対応する通信方式
- 無線 LAN のインターフェイスと対応する通信方式
- 自宅でパーソナルコンピュータをネットワークに接続しているなら、

その接続方式，無線 LAN の場合は暗号化方式

演習 11. 情報セキュリティ e-Learning の修了

本学が提供している情報セキュリティ e-Learning を修了してください。

演習 12. パーソナルコンピュータの学内ネットワークへの接続

あなたがパーソナルコンピュータやスマートフォンをお持ちなら以下の実施手順を確認してください。

- パーソナルコンピュータを学内の無線 LAN (KUINS Air)に接続する。

演習 13. Web の暗号化と証明書の確認

本学があなたに提供している KULASIS や PandA などの Web システムで通信が暗号化されており，適切な証明書を有していることを確認してください。

演習 14. 適切な電子メールの利用

授業などで担当教員に電子メールで連絡する際に，適切なメッセージの構成について述べてください。

演習 15. ファイルの転送

以下の機器の間でのファイル転送について，方法を検討してください。複数の方法があればその得失を考えてください。ただし，転送元，転送先の者以外にはそのファイルへのアクセスは許さないことを条件とします。

- あなたのパーソナルコンピュータとスマートフォンとの間でファイルなどの転送
- 教育用コンピュータシステムの PC 端末とあなたのパーソナルコンピュータの間でのファイルなどの転送
- グループで活動する際のグループ内でのファイルの共有

4. 知的生産について

4.1 本章の目的

本章ではレポート作成などの活動を知的生産ととらえます。この章では後につづく具体的なスキルを位置付ける前提として

- 知的生産とはどういうことか
- 具体的にどのような活動が求められているのか
- 知的生産を効果的、効率的に行うために ICT がどのように活用されるのか
- 知的生産やそのための ICT 活用スキルを身に着ける上で考えておくべきことは何か

を知って頂くことです。

4.2 知的生産とは

「知的生産」とは民族学者だった梅棹忠夫氏の造語です。同氏の著書「知的生産の技術」では以下のように説明されています。

「ここで知的生産とよんでいるのは、人間の知的活動が、なにかあたらしい情報の生産にむけられているような場合である、とかんがえていいであろう。この場合、情報というのは、なんでもいい。知恵、思想、かんがえ、報道、叙述、そのほか、十分ひろく解釈しておいていい。」（梅棹 忠夫：知的生産の技術（岩波新書）岩波書店（1969））⁴

すなわち、現代社会では多数の人は主に「知的生産」に携わっているのです。

「生産」という言葉で連想されるのは農林水産業や製造業で食料、エネ

⁴この本は名著ですが、書かれたのは 1969 年です。パーソナルコンピュータもインターネットも携帯電話もまだない時代に書かれています。著者は民族学（文化人類学）が専門で、フィールドでの調査をまとめて論文、著書にするというのが仕事の形態です。同書では著者の専門での仕事の仕方を背景に「知的生産」について書かれています。

4 知的生産について

ルギー、住居、衣類、機械などを生産することでしょう。しかし現代の社会では多くの人は直接、物的生産に関わりません。図 35 は職業別就業者の構成割合の推移です。この図を見ると農作業などに関わっている就業者はすでにかなり少なく、製造業などで直接、生産工程などに関わっている就業者も減少傾向にあることが分かります。これに対して赤色で囲った部分が知的生産、例えばオフィス業務、専門的業務、販売業務などに関わっている就業者に相当し、すでに就業者の中で大きな割合を占めるとともに、増加しつつあることが分かります。

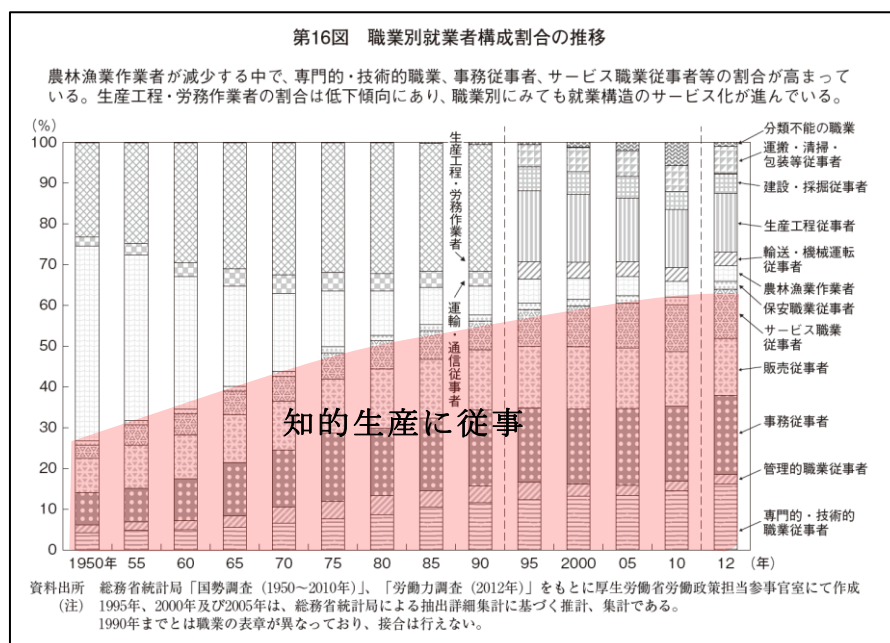


図 35 職業別就業者の構成割合の推移

平成 25 年版 労働経済の分析＜要約版＞－構造変化の中での雇用・人材と働き方－，図 1 6 に加筆

<http://www.mhlw.go.jp/wp/hakusyo/roudou/13/dl/13-2-4.pdf> (2017/1/5 アクセス)

4.3 学園祭の活動と知的生産

学生生活に引きつけて知的生産について具体的に考えてみるために学園祭に模擬店を出店することを考えてみましょう。学園祭で模擬店を出すにあたって必要な作業などを書き出してみるとさまざまなものがありま

す。

- 模擬店を出す仲間作り
- 学園祭の実行委員会の出店条件などの確認
- 何を出店するかの検討
- 企画書の作成と提出，許可をもらう
- 機材類の手配，
- 安全衛生管理の理解，実施体制の検討
- スタッフの役割分担
- 予算，チケットの価格決定，印刷，販売
- 実施
- 決算，意見集約と反省
- 報告書の提出

ずいぶん，多くの作業が伴いますが，このうち，太字の箇所は，実際に「もの」を動かしたり，加工したりする作業は殆どなく，打ち合わせをしたり，調査や手配をしたり，売り上げや支出を想定して予算など立てたり，文書を作成したり，といったもので，ほとんど「情報」だけを扱う「知的生産」になります。

知的生産をいかに効果的（役に立つ），効率的（手間をかけない）に行うかが重要になりますし，知的生産が情報を扱うものですから，そのためにいかにパーソナルコンピュータやインターネットなどの情報通信技術（ICT）を使うかが重要になるのです。

4.4 大学生活と知的生産

先の例では学園祭での模擬店出店で多くの知的生産活動が伴うことを見ましたが，大学生活では様々な知的生産活動を行わなければなりません。

- その代表例は**授業で課されるレポート**であるとか，**卒業論文**などの作成です。

このほかにも

- さまざまな目的での**文献調査**や**アンケート調査**などを行い，その**報告書**をまとめることがあります。
- イベントやプロジェクトなどに関わると

4 知的生産について

- その実施を主張するための**企画書**の作成
- 具体的に実施するための**実施計画書**の作成
- イベントでの**アンケート調査**のための質問紙の設計
- 活動を広く伝えるための**広報物**の作成
- 活動を行った後の**実施報告書**の作成
- **実施報告の評価**

などが行われるでしょう

- またソフトウェア（コンピュータのプログラム）や情報システムなどを作ると
 - システムが何を実現すべきかを書いた仕様書
 - システムがどう設計されているかを書いた設計書
 - プログラムそのもの
 - 操作マニュアル
 - Q & A 集,

などの作成が必要になります。

最近では文書などの報告書とともに、

- スライドを作成して**プレゼンテーション**を行うことが多く求められるようになっていきます。また、これらの活動をチームで行う場合には

- 会議資料や議事録の作成が求められますし、

- 会議を上手に運用することや、
- プロジェクトの進捗を管理するプロジェクトマネジメント的手法の適用も必要になります。

4.5 知的生産を構成する活動

知的生産を構成する活動にはいくつかあります。

- 調査
 - 文献調査
 - フィールド調査、アンケートやインタビューあるいは観察
 - 実験

4 知的生産について

- コンピュータシミュレーション
- 資料の整理・分析・発想
 - ブレインストーミングや KJ 法などの発想法の活用
 - 統計的分析やテキストマイニングなどのデータからの知識獲得
- 問題解決策の考案
- レポート作成
- スライド作成，ポスター作成とプレゼンテーション
- 会議の運用と議事録作成
- プロジェクトの管理

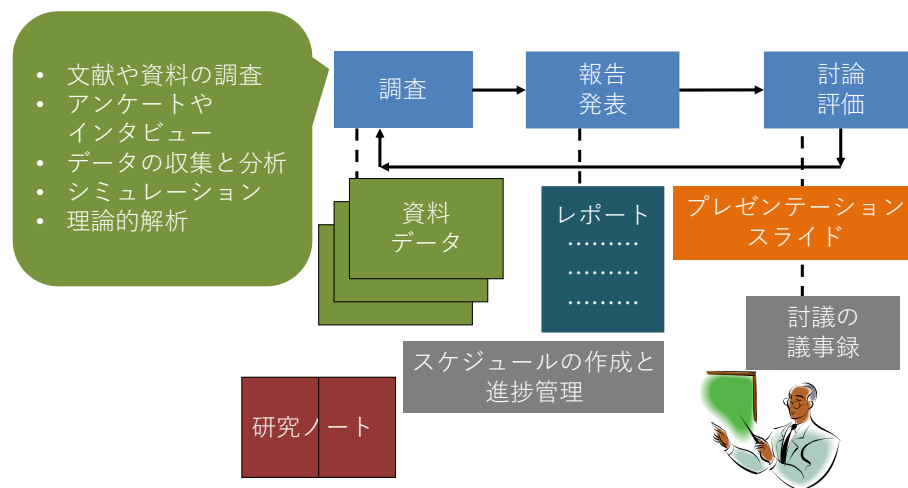


図 36 知的生産のプロセス

4.6 知的生産と ICT 活用

知的生産は主に情報を扱う仕事ですから、そのために ICT をいかに活用できるかは重要なことでしょう。授業での課題についてグループでレポートを作成し、プレゼンテーションすることを考えてみましょう。以下のように ICT を活用するでしょうし、そのためのスキルが重要であることが分かります。

- **文献調査：** 図書館での蔵書検索やデータベース検索を活用した資料調査や Web での情報検索
- 政府による統計などの **データ活用：** 政府や自治体などはさまざまな統計を Web で公開しています。そのデータを表計算ソフトを用い

て分析したり，グラフに表したりすることが求められます。

- **資料整理とノート作り**：文献などの資料，データをコンピュータ上のファイルとして整理したり，検索したりすることが求められます。また，読んだ文献の内容などは出典を記載するための書誌情報とともにノートのまとめておくことが必要でしょう。
- コンピュータを用いた**シミュレーション**：表計算ソフトを用いたり，自分自身でプログラミングしたり，専用のソフトウェアを用いたりするなどしてシミュレーションのための数値計算を行います。
- **図表の作成**：データ分析やシミュレーションの結果を要約した表にまとめたり，グラフで表現することのほか，組織の構成などを図（ダイヤグラム）にしたりします。
- **レポートの作成**
- **スライド作成とプレゼンテーション**
- **グループでの活動**：情報共有，連絡，ミーティングと議事録の作成，レポートやプレゼンテーションスライドの校閲

4.7 知的生産スキルの獲得と学習の方略

4.7.1 スキル獲得の視点

以下の視点で知的生産スキルを獲得してください。

- あなたの知的生産活動は効果的な成果物を得ているでしょうか
- あなたの知的生産活動は効率的に行えているでしょうか
- あなたは知的生産のためのスキル獲得を継続的行えているでしょうか

4.7.2 目標の明確化

知的生産のために ICT スキルを獲得するに際しては目標を明確に意識する必要があります。すなわち

- 何（What）をできるようになればいいか，成果物としてどのようなものを生み出すことを求められているかを意識します。
- どのようにして（How）行うか。求められる成果を達成するために

ICT をどのように使えばいいかを意識します。

参考文献

- [4] 梅棹 忠夫：知的生産の技術（岩波新書）岩波書店（1969）

5. 学術情報の探索

5.1 本章の目的

この章では、前章で学習した大学における知的生産に不可欠な学術情報に関して、1) 探索方法、2) 評価方法、3) 適切な活用方法の3点を学習します。大学における知的生産の中で、この章で想定しているのは、授業レポートや卒業論文など、皆さんの学習や研究の到達度を文章で表現する場合です。具体的な内容は以下のようになっています。

- 学術情報とは何かを理解する
- 検索のための概念整理のやり方を身につける
- 引用の仕方と参考文献リストの作成方法を身につける

これらを修得した上で、実習を通してオンライン文献検索の基礎を身につけましょう。

5.2 学術情報と図書館

5.2.1 学術情報とは何か

学術情報の特徴を説明する前に、情報化時代に大学で学習する内容について少し考えてみたいと思います。私たちの普段の生活では、分からないことがあればネットで検索しておおむね概要をつかむという習慣が定着しています。ネット検索では、ずらっと並んだ検索結果のそれぞれが書かれている文脈を考えるよりは、同じ単語を使っている、少しずつ違う文脈内書かれた内容を、なんとなくつなぎ合わせて考えることが多いかもしれません。一方で、大学での学習や研究をすすめる上では、文脈（コンテキスト）であるとか、各学問分野における枠組み（フレームワーク）の中で理解し、自分の意見や発見をその中に位置づけるという作業が重視されます。

もちろん、「東南アジアにおける、エビの養殖事業をめぐる流通と環境問題」といったように、ちょっと考えるだけでも、経済学、環境学、地域研究といったいくつかの分野にまたがる研究テーマはたくさんあります。グローバル化した現代社会における事象が研究対象であればそういった分野横断型の課題の方が多いでしょう。しかし、分野を超えたテーマにアプローチする基礎体力として、まずは自分の所属する学部が対象とする学問分野の文脈と枠組みにのっとなって考える習慣を身につけ

ることが必要になります。文脈と枠組みというのは、スポーツに例えるとルールと競技会場といったところでしょうか。競泳と野球では、会場もルールも全く違いますが、どういった場で行うかが規定されていて、一定のルールに従って動き、先行事例を参考に戦い方を決めるといった点は共通しています。学問の場合でも、同じことがいえます。

このような前提に基づいて、学術情報を定義してみますと「特定の分野における、学問の進捗過程を、世の中に公開するためにつくられた情報」となり、具体的には、以下のような形態があげられます。

- 学会発表の内容
- 学術雑誌に投稿された論文
- 専門書
- 専門家によって執筆された百科事典や専門事典の項目

（百科事典といっても Wikipedia は誰でも書き換え可能なので学術情報ではありません）

そのような学術情報の集積の中で、先行事例、つまり先行研究を網羅的に検証した上で、自らの論を組み立てたり、新たな調査を行ったりするというのが大学における知的生産活動だといえます。

5.2.2 知識の定着と資料

学術情報として具体的にいくつかの種類の資料をあげましたが、知識の定着というのは、河の流れのようになっていて、多くの場合は、学会や学術雑誌で発表された最先端の研究内容が時間とともに定説化していきます。上流にあたる学会での発表や、研究論文として公開された新たな知見は、同じ分野の研究者の間で相互に評価・検証する「ピアレビュー（peer review）」というプロセスを経た上で、学術的に信頼できる情報としての評価が確立されます。その上で、学術書や専門書に掲載されることで、一般読者よりは限定されているかもしれませんが、近接分野の専門家にも知識として共有されるようになります。さらに下流に行くにしたがって、より一般的人人に使いやすい、教科書や事典といった出版物や媒体に発信先が変化していき、その過程で残った知見は一般的な知識として定着していきます。

「iPS 細胞（人工多能性幹細胞）」を例にとって考えてみましょう。「iPS 細胞（人工多能性幹細胞）」は 2006 年に、*Cell* という学術雑誌に掲載された論文記事として世に出ました。その時点では、専門家を中心としてごく限られた範囲で共有されてい

た知識にすぎませんでした。が、*Cell* に掲載した論文の著者である山中伸弥教授と、1960 年代にその先行研究にあたとされる論文を発表したイギリスのジョン・ガードン博士が、2012 年にノーベル生理学・医学賞を受賞したことにより、多くの人々に知られる用語となり、現在では「iPS 細胞（人工多能性幹細胞）」は、多数の一般書や百科事典などに掲載されています。

このような時間の経過と知識の定着の関係を、文献の種類とあわせて知っておくと、自分の知識レベルにあわせて学術情報の探索ができますので、頭の片隅にしまっておいて下さい。自分にとって全く新しいテーマに挑戦する時は、河の下流からさかのぼると、迷わずにたどり着けるはずです。

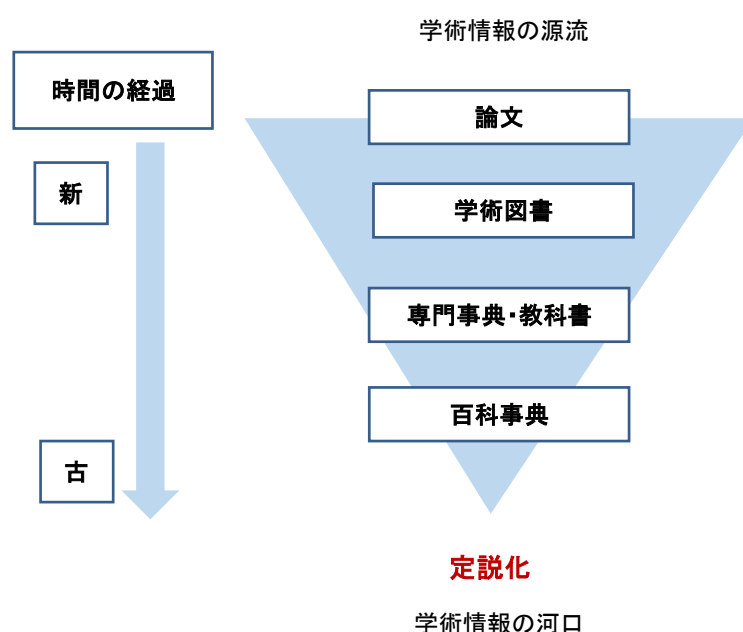


図 37 時間の経過と知識の定着

5.2.3 ネット上の学術情報と図書館が提供する学術情報の違い

さて、学術情報についてある程度理解したところで、ネット情報と図書館が提供する学術情報について簡単に整理しておきます。簡単にいうと、図書館経由で得られる学術情報は、「信頼性」が高いということです。あまたある情報の中で、図書館ではすでに定評がある学術出版物や、電子ジャーナルなどを購入し、提供しています。これは、専門家によるピアレビューを経た知識ということでもあります。一方で、ネット経由で手に入る学術情報は玉石混淆ですので、その中から本当に重要な学術情報を探し出すには、ある程度知識がないと難しいことが多いでしょう。

図書館では、以下のような「信頼性」の高い学術情報が提供されていますので、これらを活用しない手はありません。なお、これらの図書館が提供している学術情報は、大変高価な資料で、個人で購入することが難しいものも多く含まれています。

- 参考資料（辞書・地図・統計など読みものではなく、調べるための資料を指します）
- 図書
- 学術雑誌
- 電子ブック
- 電子ジャーナル
- オンライン・データベース
- 新聞データベース
- 官公庁のレポートなど
- 機関リポジトリ

なお、新聞データベースや官公庁のレポートなどは、就職活動時や就職した後の調べ物にも役にたつ資料です。「信頼性」のフィルターをかけた資料を収集している図書館という場所で調べ物をはじめる習慣をつけておくと、学術情報以外にも、質の高い情報を選ぶ力が身についていきます。

5.3 検索を行う前に

5.3.1 文献検索の種類

学術情報と図書館については理解できたと思います。問題は、図書館という「信頼性」フィルターを通した後であっても、アクセスできる情報は桁違いに多いことです。このような情報の海の中で、網羅的な先行研究の収集を行うにはどうしたらよいのでしょうか。図書館における文献探索には、大きく分けて、1) ブラウジング、2) 芋づる式、3) オンライン検索の3種類があります。3つめのオンライン検索については、次節で詳しく説明しますが、3種類の文献探索方法を上手に組み合わせることが、不可欠です。

1) ブラウジング

- 図書館の棚をみながら探すことです。
- **メリット**⇒図書館では、分類方式に従って近い分野の本が同じ棚に並んでいますので、自分の考えているテーマを、より広く捉え直すことができますし、オンライン検索ではうまく探せなかった資料に出会えることも多々あります。
- **デメリット**⇒多分野にわたるテーマの場合、どこに分類されているかが分かりにくいことがあります。また、棚にあった本だけをみて網羅的な資料収集をしたような気になってしまわないように気をつけて下さい。

2) 芋づる式

次節で説明する、先行研究の参考文献リストに挙げられている文献を探すことです。芋づる式を可能にする引用データベースというのがありますので、それに関しては章末のコラムを参照してください

メリット⇒自分の取り組んでいるテーマにおいて、重要な文献を効率良く入手できます。

デメリット⇒先行研究が言及していない重要な文献を見落としてしまうことがあります。

5.3.2 オンライン検索とキーワードの整理

次に、3つめの選択肢であるオンライン検索ですが、効果的なオンライン検索を行うために、まず必要な作業とはなんでしょうか。それは、自分が考えているテーマをキーワード化し、それらのキーワードの上位語、等位語、下位語を考えることで、検索キーワードをめぐる用語や概念を整理することです。ブラウジングや芋づる式と違い、オンライン検索の場合は、入力したキーワードによって検索結果が劇的に変わってきます。

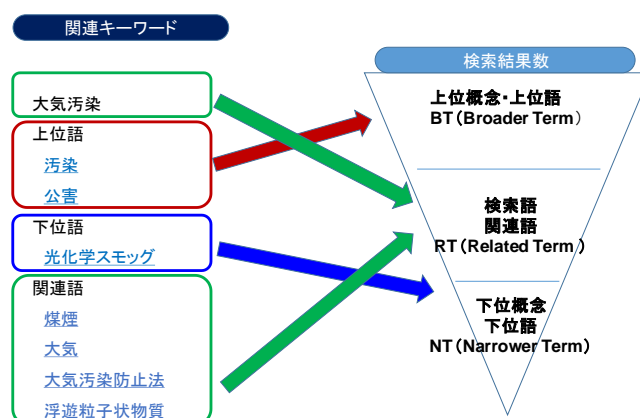


図 38 上位概念・下位概念

「現代社会におけるマイノリティと法」というレポート課題が出たとしましょう。現代社会とは、どの範囲を指すのか、どのようなマイノリティを取り上げたいのか、どのような法を考えたいのか、を整理していく中で、「同性婚」というテーマについて調べようと考えたとします。ところが、「同性婚」というキーワードだけで検索していても、なかなか思うような文献に出会わないかもしれません。そういった時は、検索しているキーワードの上位概念や、関連語を考えてみましょう。また逆に、自分の思っている検索語では、ヒットする文献の数が多すぎてどれを読んだらよいのか分からないということあります。そういう場合は、自分が考えた検索語の下位語を考えてみましょう。

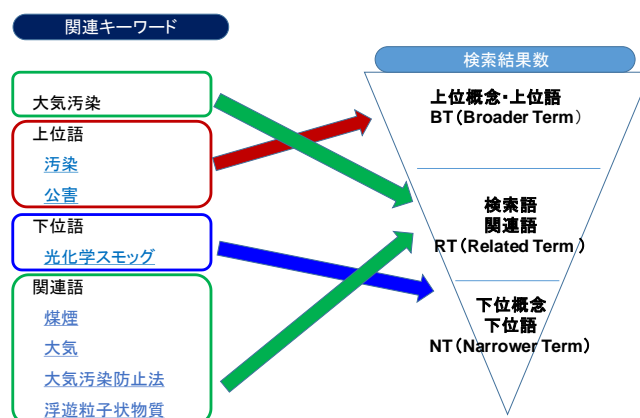
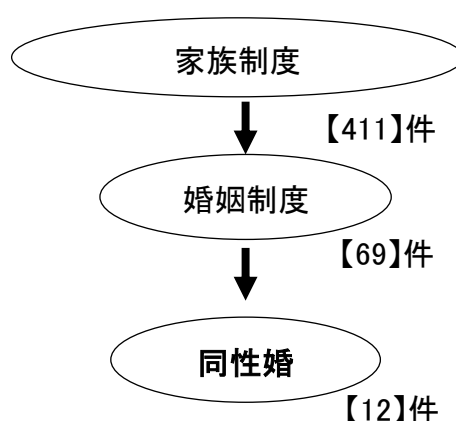


図 38 の左側は、「大気汚染」というキーワードで、国立国会図書館の文献データベース『国立国会図書館サーチ』（URL: <http://iss.ndl.go.jp/>、2016 年 10 月 3 日時点）を検索した場合に提示される、上位語・下位語・関連語です。「大気汚染」の上位語である「公害」で検索をするとしたら、「大気汚染」を含んだ、さまざまな「公害」に関する文献がヒットしますので、検索結果数は増加しますし、下位語の「光化学スモッグ」で検索した場合、「大気汚染」の中でも「光化学スモッグ」に言及してい

る文献しかヒットしないので、検索結果数は減ります。

このように、文献の探索する際には、検索概念のレベルを行き来しながら、模索することが必要になります。オンライン検索を行っているうちに、他にも有効なキーワードが見つかることは多々ありますので、どの検索ワードによって、どのような文献を入手できたか、文献とキーワードを精査することで、自分が必要な情報を見つけていきましょう。このようにして入手した文献の中で、同じテーマを扱っている論文が高い頻度で引用している文献は、学術情報の上流に位置する文献、つまり「基礎文献」である確率が高いですので、最新の研究成果と基礎文献の双方を活用する習慣をつけましょう。



【 】内は、2016年10月3日のKuline検索結果

図 39 検索語と検索結果数

5.4 参考文献リストの読み方・書き方

5.4.1 引用と参考文献リストの重要性

もう一つ、文献検索をはじめる前に知っておく必要があるのが、参考文献リストの読み方と書き方です。参考文献リストとは、論文や本の最後に掲載されており、著者が引用した文献を一覧にしたものです。これを見ることによって、著者がどのような先行研究を参考にし、その上で、どのような新しい発見を提示しているのかが一目瞭然となります。前節で、芋づる式の文献探索方法について説明しましたが、レポートや論文を書く作業は、参考文献リストを読むことから始まり、参考文献リストを書くことで終わるといっても過言ではありません。なお、参考文献を示すことの重要性については、科学技術振興機構（2011）の『参考文献の役割と書き方』で、以下の5つが挙げられています。

1. 自身の論文の新規性、独創性、信頼性の明確化
2. 先行する著者（先人・先輩）に対する敬意
3. 出典の明示
4. 読者に対する情報提供
5. 著作権法の遵守

引用と参考文献リストを読んだり、作ったりする上で、もう一つ重要なのが、決められた書式に従うということです。フォーマットは、専門分野によって違いますので、レポートを提出する場合は、事前に先生と確認しましょう。

引用・参考文献の代表的な書式

- APA（American Psychological Association：アメリカ心理学会）
【心理学をはじめ、社会科学分野全般】
- Chicago 【人文学（歴史・語学）・社会科学分野】
- MLA（Modern Language Association of America 米国現代語学文学協会）
【人文学（文学・言語学・哲学）】
- IEEE（The Institute of Electrical and Electronics Engineers：電気電子学会）
【電気・通信・電子・情報工学分野】
- SIST（Standards for Information of Science and Technology：科学技術情報流通技術基準）

本文で引用した資料は、必ず参考文献リストに挙げる必要があります。書式の多くが、引用と参考文献のどちらも指定していますので、その場合は、どちらも同じ書式に従って記載しましょう。

英文の書式による記述

- APA
（本文中）（Klaus,1981）
（参考文献リスト）B. Klaus. (1981). Robot Vision. Cambridge, MA: Little, MIT Press.
- ◆ IEEE
（本文中）as known by Klaus [4]
（参考文献リスト）[4] B. Klaus, Robot Vision. Cambridge, MA: MIT Press, 1981.

和文雑誌の書式による記述例

- 『社会学評論』
(本文中)(西川編 2001)
(参考文献リスト) 西川潤編, 2001, 『アジアの内発的发展』 藤原書店。
- 『東南アジア—歴史と文化—』
(本文中) [生田・松澤 2000]
(参考文献リスト) 生田真人・松澤俊雄編. 2000. 『アジアの大都市 3 クアラル
ンプール, シンガポール』 日本評論社。

また、文献リストを作成する場合は、同じ書式でも、文献の種類によって形式が違います。特に指定がない場合は、SIST 書式を使用しましょう。

SIST による文献リスト

(以下の例では、便宜上分かりやすい箇所で改行しています)

- 図書の場合
 - ◆ 単著図書の全体を参考にした場合
山崎茂明. 論文投稿のインフォマティクス. 中外医学社, 2003, 167p.
著者名. 書名. (版表示,) 出版者, 出版年, 総ページ数.
 - ◆ 図書の一部を参考にした場合
山崎茂明. “国際ルールへの理解”. 論文投稿のインフォマティクス.
東京, 中外医学社, 2003, p.71-101.
著者名.“章の見出”. 書名. (版表示.)
出版地, 出版者, 出版年, 始頁-終頁.
- 雑誌論文の場合
 - ◆ 著者が1名の和文雑誌記事を参考にした場合
根本信乃. 寿命関連遺伝子と「老化の代謝説」. 実験医学.
2003, vol.24, no.11, p.1624-1629.
著者名. 論文名. 雑誌名.
出版年, 巻数, 号数, 始頁-終頁.
 - ◆ 著者が複数名の和文オンライン・ジャーナルを参考にした場合
荒川正幹ほか. Hopfield Neural Network を用いた新しい分子重ね合わせ手法の
3D-QSAR への応用. Journal of Computer Aided Chemistry, 2002,
vol.3, p.63-72. <http://joi.jlc.jst.go.jp/JST.JSTAGE/jcac/3.63>, (参照 2016-12-03).
著者名. 論文名.

誌名，出版年，
巻数，（号数，）始頁-終頁．入手先，（入手日付）．

● ウェブページ内の記事の場合

中央教育審議会. “教育振興基本計画について－「教育立国」の実現に向けて－（答申）”. 文部科学省. 2008-04-18.

http://www.mext.go.jp/b_menu/shingi/chukyo/chukyo0/taoushin/08042205.htm, (参照 2008-08-13).

著者名. “ウェブページの題名”. ウェブサイトの名称. 更新日付.
入手先, (入手日付)

5.5 検索の実際

では、実際に図書館が提供している学術情報を検索してみましょう。その際、入り口となるのが Kuline (クライン) です。Kuline では、5.2.3 で挙げた、京都大学内の学術情報の検索や、代表的なデータベースへのリンクが提供されています。以下に Kuline へのリンクと、Kuline からアクセスできる、いくつか代表的なデータベースを挙げておきます。なお、京都大学がお金を払って契約しているデータベースの多くは、「同時アクセス数」が限定されています。データベースの使用後は、速やかにログアウトして下さい。

- | | |
|----------------------|---|
| ● Kuline | 京都大学蔵書検索
【 http://kuline.kulib.kyoto-u.ac.jp/ 】 |
| ● Cinii Articles | 日本語論文検索（全分野） |
| ● JapanKnowledge Lib | 多数の辞書や百科事典の検索 |
| ● 医中誌 Web | 医学系論文検索（日本語） |
| ● Pubmed | 医学系論文検索（英語） |
| ● Web of Science | 英語論文検索（全分野） |
| ● 聞蔵Ⅱビジュアル | 朝日新聞・AERA・知恵蔵など |

5.6 困ったら図書館に

本章では、学術情報の探し方と扱いについて概観してきました。このようなステップを経て探しても、学内で探している資料の所蔵がない場合は、図書館ネットワークを経由して、他大学からコピーや現物を取り寄せることができます。その他にも、分からないことがあれば、まずは図書館に相談してみてください。なお、附属図書館では、1階ラーニングコモンズにて、平日 13 時～19 時の間、大学院生のスタッフに学術情報の探し方やレポートの書き方などの相談をすることができる「学習

サポートデスク」が設置されています。

参考文献

文献探索についてもっと知りたい方は、以下の図書[5][6]もおおすすめです。

- [5] 井上眞琴. 図書館に訊け. ちくま書房, 2004, 256p.
- [6] 市古みどり編. アカデミック・スキルズ 資料検索入門—レポート・論文をかくために. 慶應義塾大学出版会, 2014, 151 p.
- [7] 独立行政法人科学技術振興機構. 参考文献の役割と書き方. 独立行政法人科学技術振興機構, 2011, 24p. jipsti.jst.go.jp/sist/pdf/SIST_booklet2011.pdf, (参照 2016-10-10).

Great Idea! ガーフィールド博士と引用文献データベース

現在、国際的な大学評価で使われている指標の一つに、評価対象の大学に所属する教員が書いた論文が、どのような雑誌に掲載され、どの程度の頻度で引用されているか（被引用件数）といった論文の影響力を数値化し、被引用文献から引用元をたどることができる引用文献データベースがあります。

つまり、①ひとつの論文からその研究の上流にある論文をたどり、源流にたどり着くことができますし、②その逆にある論文から、その後の研究がどのように展開していったかを調べるができるデータベースです。①の使い方をすると、その分野における「基礎文献」が分かります。

このような引用文献データベースの代表的なものに、Web of Science があります。Web of Science のはじまりは、*Science Citation Index* という冊子で、1961年にはじめて出版されました。この冊子を出版したのが、当時 30 代だったユージン・ガーフィールド（Eugene Garfield）です。ガーフィールドは、すべての論文は、それ以前に書かれた別の論文を引用・参照していることから、引用索引を作り、論文間の関係性を明らかにすることで、科学の構造化を実現できるということに気がつき、1955 年に論文で発表しました。その後、全米の州裁判所・連邦裁判所の判例を索引から引き、参照することができる冊子、*Shepard's Citation* を参考に出版したのが、*Science Citation Index* でした。現在では、自然科学のみならず、社会科学分野の論文も集積され、Web of Science というオンライン・データベースで提供されています。

Eugene Garfield. Citation Indexes for Science- New Dimension in Documentation through Association of Ideas. SCIENCE. 1955, vol. 122, no. 3159, p,108-111.

6. 表計算ソフトによるデータ処理（1）

6.1 学習の目的

- アカデミックな活動における表計算ソフトの利用目的を理解すること。
- 表計算ソフトの基本構造としてセルの内容と他のセルの値の参照法について学びます。
- 表計算ソフトへのデータの入力方法について、表の構成法と入力の制限法を学びます。
- 表計算ソフトに入力されたデータの特定の項目に注目した並べ替え方法を学びます。
- 表計算ソフトに入力されたデータの集計法について学びます。
- ピボットテーブルを用いたデータのクロス集計法について学びます。

6.2 表計算ソフトの利用目的

アカデミックな ICT スキルとして表計算ソフトを利用する目的についてここでは以下の3つを考えます。

- 大量のデータを適切に入力し、集計などを行って分析すること。
- 表計算ソフトを利用してシミュレーションを行うこと。
- 集計やシミュレーションによって得られたデータをグラフに表現すること。

6.3 表計算ソフトの基本構造

6.3.1 セルの内容

表計算ソフトは「セル」が縦横に碁盤目状の並んだ「シート」で構成されています。横方向の並びを「行」、縦方向の並びを「列」と呼びます。

6 表計算ソフトによるデータ処理（1）

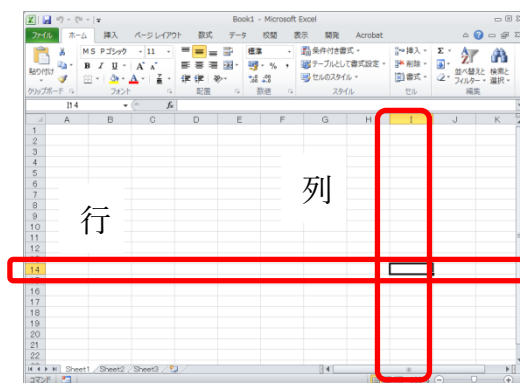


図 40 表計算ソフトのワークシート，行，列，セル

セルには「値」として数値や文字列が入力できるほか，他のセルの値を使った計算式を入力できます。図 41 の例ではセル A6 には数値ではなく式が入力されています。

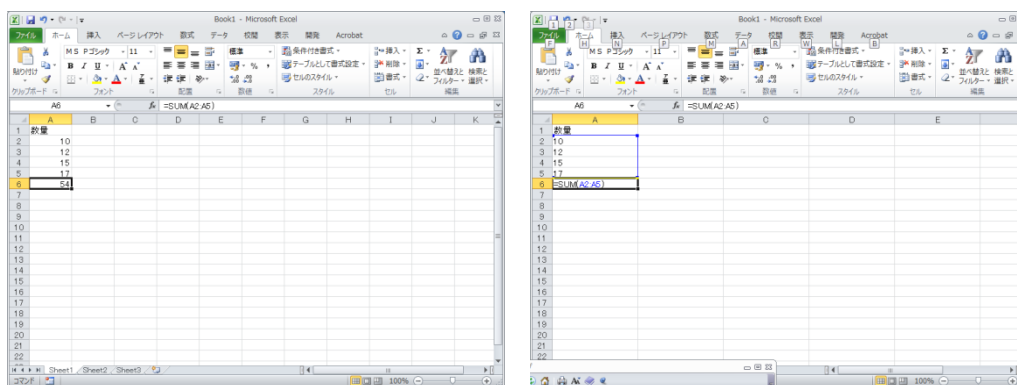


図 41 セルの内容

（右側の図は意図的に式が見える表示をしています。）

「ファイル」→「オプション」→「詳細設定」→「シートで作業する時の表示設定」の中で「計算結果の代わりに数式をセルに表示する」をチェック）

6.3.2 他のセルの値の参照

1) セルの値の参照

セルに式を書く際に他のセルの値は列の名前と行の番号を表記することで参照できます。例えば

$$=A2+A3+A4+A5$$

とすれば A 列の 2 行～5 行のセルの値の和を計算できます。

2) 矩形領域の指定

合計を計算する sum 関数などで矩形の領域の複数のセルを指定する際には例えば

=sum(A2:A5)

のように左上と右下のセルをコロン「:」でつないで表記します。

3) 相対参照と絶対参照

式を他のセルにコピーした場合、式の中で参照しているセルは式のあるセルからの相対的な位置を保存してコピーされます。例えば A6 のセルにある

=sum(A2:A5)

を隣の B6 にコピーするとセルの内容は

=sum(B2:B5)

となります。これを**相対参照**と言います。

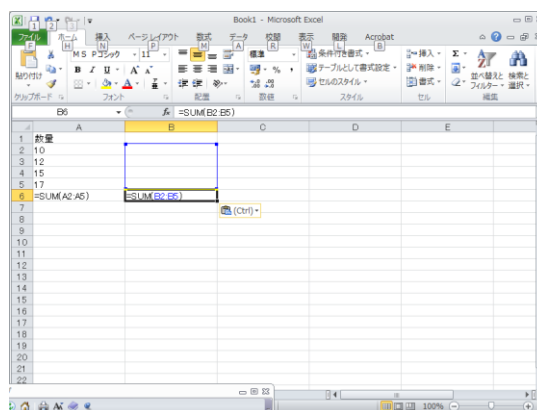


図 42 領域の参照

これに対して、コピー先に関わらず特定のセル（行や列）を参照したい場合には行や列の名前の前に \$ 記号をつけます。これを**絶対参照**と呼びます。下の例では B2 のセルに A2 のセルの 2 割増しを計算するため定数 1.2 をセル C2 に入れて、以下のように**絶対参照**で

=A2*\$C\$2

と表記しています。これを B3~B5 にコピーすると「A2」に相当する部分は相対参照ですのでそれぞれ A3~A5 に変わりますが、C2 については同じセルを参照します。

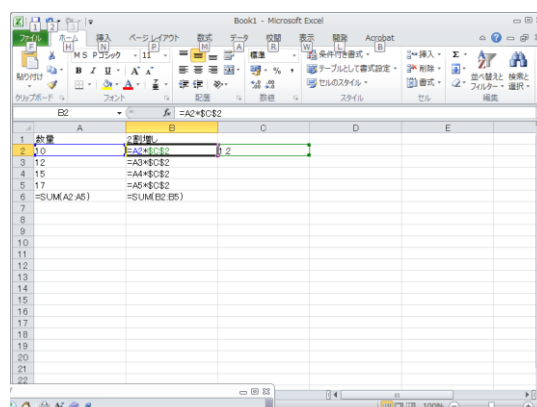
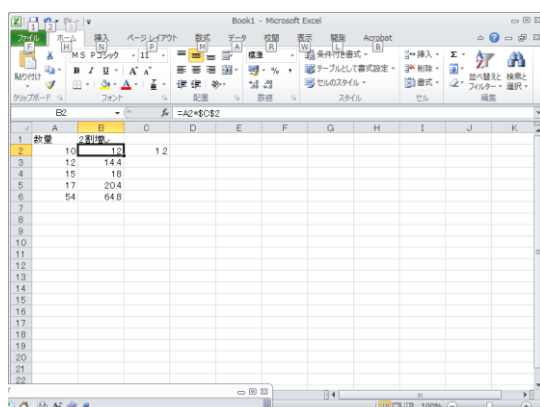


図 43 相対参照と絶対参照

Great Idea! 表計算ソフトの開発

Dan Bricklin と Bob Frankston は 1979 年に表の中に数値だけでなく、表内の他のデータを参照する計算式を書けるようにし、自動的に計算するという表計算ソフト VisiCalc を Apple II コンピュータ用に開発しました。これは Apple II コンピュータのキラーアプリケーションとなりました。現在の Excel などでも用いられている「A1」などのセルの参照表記は当時の VisiCalc の実装から継承されています。このアイデアは大量のデータの集計を誰もが行えるようにする画期的なものです。

演習 16. 相対参照と絶対参照

本節で示した例を実際に Excel で作成し、式の中で他のセルの値を参照することと、その際の相対参照、絶対参照の機能の違いを確認してください。

6.4 データの入力方法

6.4.1 データの収集と分析

ここでは調査などでデータを集め、それを表計算ソフトに入力し、分析することを考えます。その際、以下のような作業が必要になります。

- 調査の概要など必要な事項を記録すること。
- データを表計算ソフトに入力すること。その際、さらに以下の点に注意をします。
 - 後のデータ処理が容易な**表の構成**を用いること。
 - 誤りが極力少なくなるような**データの入力の工夫**をすること。
 - 入力されたデータが正しいかどうかを**確認**すること。

6.4.2 表の構成

1) シートの使い方

- 調査の概要なども Excel のシートに記載するようにします。Sheet1 を用いることにし、シートの名前を適宜変更します。
- 調査で得られるデータ Sheet2 に記載するようにします。同じくシートの名前を適宜変更します。
- 集計結果などは Sheet3 以降を使うようにします。

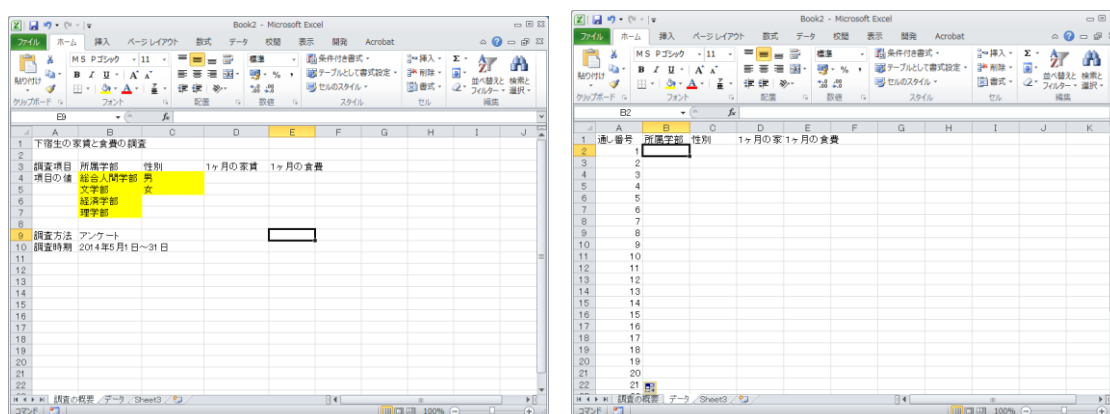


図 44 シートの使い方

2) データ表の構成

アンケート調査など、多くのデータは1件のデータが複数の項目で構成され、調査結果は多くの件数のデータで構成されます。これをシートに記載するには以下の方法を標準に行います。

6 表計算ソフトによるデータ処理（1）

- 1 件のデータを 1 行で構成します。
- 表の 1 行目には**項目の見出し**を書きます。
- 1 列目は通し番号用に使うと便利です。
- データは 2 行目以降に入力して行きます。

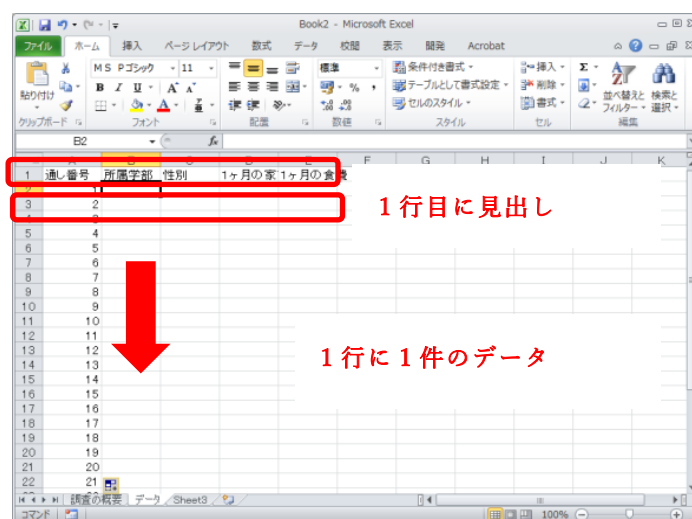


図 45 表の構成の基本

6.4.3 入力の制限

データの入力はいへん労力を要し、なおかつ人が行うため**必ず誤りが発生する**と考えることが重要です。このため作業効率を高める工夫と誤りを極力なくする作業手順が重要です。

1) ウィンドウ枠の固定

見出しや通し番号を常に見えるようにしておけばデータの入力や確認が容易になります。このために「ウィンドウ枠の固定」という操作を行います。見出しや通し番号以外のデータの左上のセル（セル B2）を選んだ状態で、「表示」タブのリボンから「ウィンドウ枠の固定」を選びます。

6 表計算ソフトによるデータ処理（1）

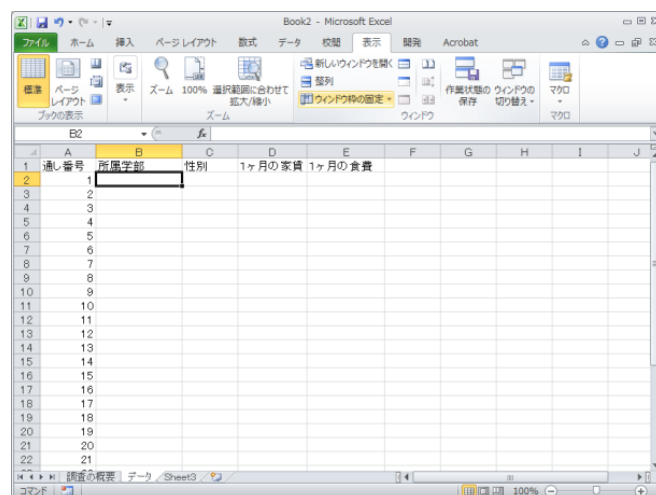


図 46 ウィンドウ枠の固定

2) データ入力規則の設定

データの内容が例えば「学部」とか「性別」などのように**特定の文字列**に限られている場合，これを**直接タイプ入力してもらったと，表記が揺らぐ**などして多くの誤りが発生します。これを予め定められたものから選べるようにするには以下のようにデータの入力規則を設定します。

- 入力候補の文字列を例えば調査の概要のシート(Sheet1)に作成します。
- データ用のシートで，入力対象となるセルを選びます。
- データタブを選んで，リボンから「データの入力規則」を選び，「設定」として「リスト」を選びます。
- 「元の値」として，先に設定した候補の文字列を選びます。調査の概要のシートに移動し，候補文字列の先頭を選び，Shift キーを押したまま，最後の候補文字列を選べば，候補文字列を設定できます。
- 入力規則を設定したセルを下方に必要な分だけコピーします。

6 表計算ソフトによるデータ処理（1）

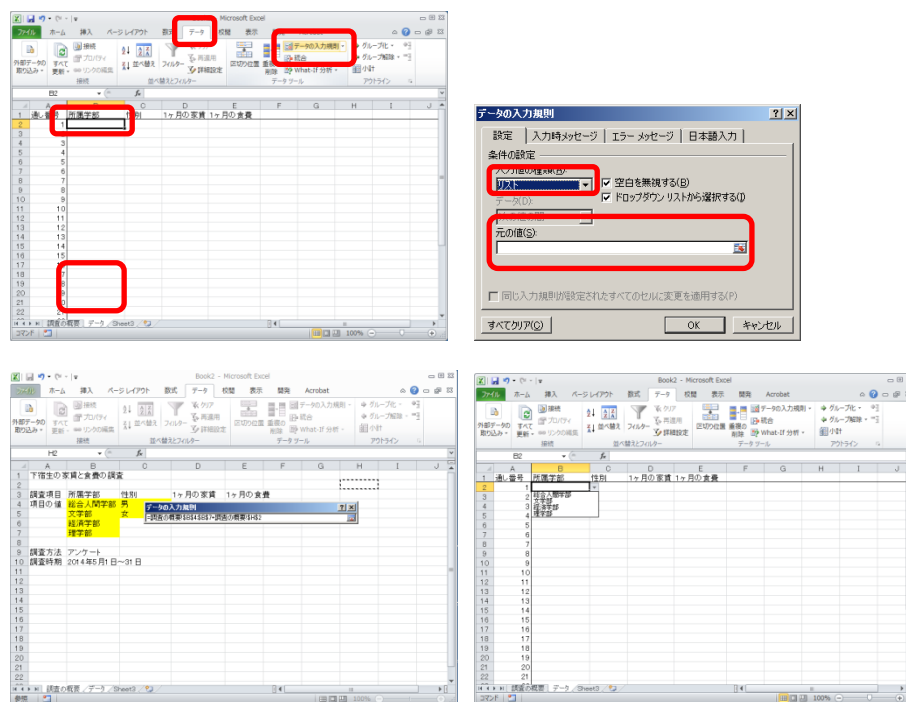


図 47 入力の制限方法

2人で作業できる場合などは、

- 1人が読み上げと入力の確認、
- もう1人が表計算ソフトの操作を担当する

と効果的です。

データの入力が終わったら、必ず元のデータが正しく入力されているか照合するようにします。

演習 17. データの作成

表 4 に示したデータを本節で示した手順に沿ってシートとして構成し、データ入力規則の設定などして入力してください。ただし学部は総合人間学部、文学部、経済学部、理学部が現れます。

表 4 演習用のサンプルデータ（架空のものです）

通し番号	学部	性別	1ヶ月の家賃	1ヶ月の食費
1	文学部	女	41000	16000
2	文学部	女	39000	13000
3	経済学部	男	22000	15000
4	経済学部	男	48000	14000
5	経済学部	男	26000	15000
6	文学部	女	51000	19000
7	文学部	女	49000	21000
8	総合人間学部	男	28000	19000
9	文学部	女	40000	11000
10	理学部	女	45000	10000
11	理学部	男	37000	11000
12	総合人間学部	女	32000	14000
13	理学部	女	32000	10000
14	経済学部	男	30000	12000
15	理学部	男	46000	17000
16	経済学部	男	29000	13000
17	総合人間学部	女	51000	27000
18	経済学部	男	34000	14000
19	文学部	女	23000	14000
20	経済学部	男	20000	14000
21	文学部	男	45000	17000
22	文学部	男	21000	16000
23	経済学部	女	50000	14000
24	理学部	男	48000	25000
25	経済学部	女	51000	26000
26	文学部	女	24000	23000
27	経済学部	男	24000	24000
28	文学部	女	26000	22000
29	経済学部	男	32000	24000
30	文学部	女	45000	23000

6.5 データの並べ替え

データを使った分析では特定の項目に注目して、その値でデータを並べ替えたいことが少なくありません。

- データの並べ替えのことは「整列」や「ソート(sort)」とも呼ばれます。
- データの値が小さいものから大きくなる順に並べることを「昇順」と言います。
- 逆に大きいものから小さくなる順に並べることを「降順」と言います。
- 文字列は文字を表すコードの順で並べ替えられます。漢字の場合は読みの項目などを作ってそれで並べ替える必要があります。

前節の演習 17 で作ったデータを並べ替えてみましょう。

- データのあるセルをどれか選びます。
- データタブを選び、リボンの中の「並べ替え」を選びます。
- 最優先キーを並べ替えたい項目に、順序を昇順や降順に選びます。
- 先頭行をデータの見出しとして使用する、が指定されていることを確認します。
- OK を選ぶと並べ替えが完了します。

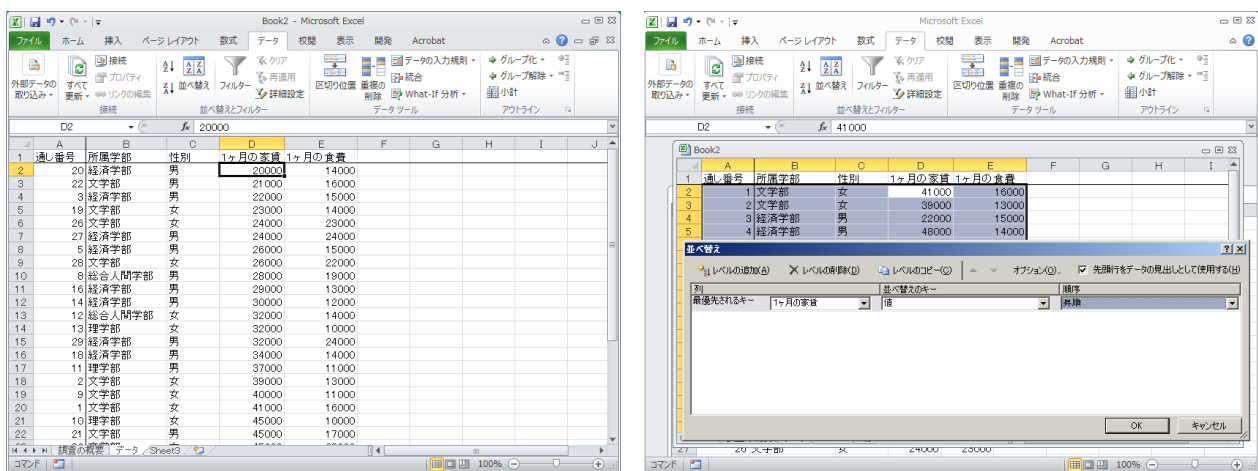


図 48 データの並べ替え

演習 18. データの並べ替え

演習 17 で入力したデータについて家賃や食費で並べ替えを行い、それぞれ高いモノから 3 件、低いものから 3 件の値を書きだしてください。

家賃の高いもの 3 件			
家賃の低いもの 3 件			
食費の高いもの 3 件			
食費の低いもの 3 件			

6.6 データの集計

データを集計するために Excel にはいくつかの関数が用意されています。

6.6.1 個数を数える

表 5 個数を数える関数

関数名	意味	例
count	範囲内の数値の個数を数える	count(A2:A31)
counta	範囲内の空白でないセルの数を数える	counta(A2:A31)
countif	範囲内で条件を満たすセルの個数を数える	countif(B2:B31,"工学部") countif(D2:D31,">20000")
countifs	範囲内で複数の条件を満たすセルの個数を数える	countifs(B2:B31,"工学部", D2:D31,">20000")

countifs 関数では異なる列の範囲に対して条件を設定して数えることができます。例えば上の表に示した例では

- B 列の 2 ～31 行の値が“工学部”で、
 - D 列の 2～31 行の値が 20000 より大きい、
- 行の数を数えます。

6.6.2 合計を求める

表 6 合計を求める関数

関数名	意味	例
sum	範囲内の数値の合計を求める	sum(D2:D31)
sumif	範囲内の条件を満たすセルの値の合計を求める	sumif(D2:D31,">20000") sumif(B2:B31,"工学部",D2:D31)
sumifs	範囲内で複数の条件を満たすセルの値の合計を求める	sumifs(D2:D31,B2:B31,"工学部", C2:C31,"男")

sumif の 2 つ目の例は条件検索をする範囲が B 列（B2:B31）ですが、実際に合計するのは D 列という場合の書き方です。

6 表計算ソフトによるデータ処理（1）

sumifs 関数は複数の条件を書けますが、最初に合計する範囲指定して、後で条件を検索する範囲と検索条件を対にして必要な数だけ書きます。

6.6.3 平均を求める

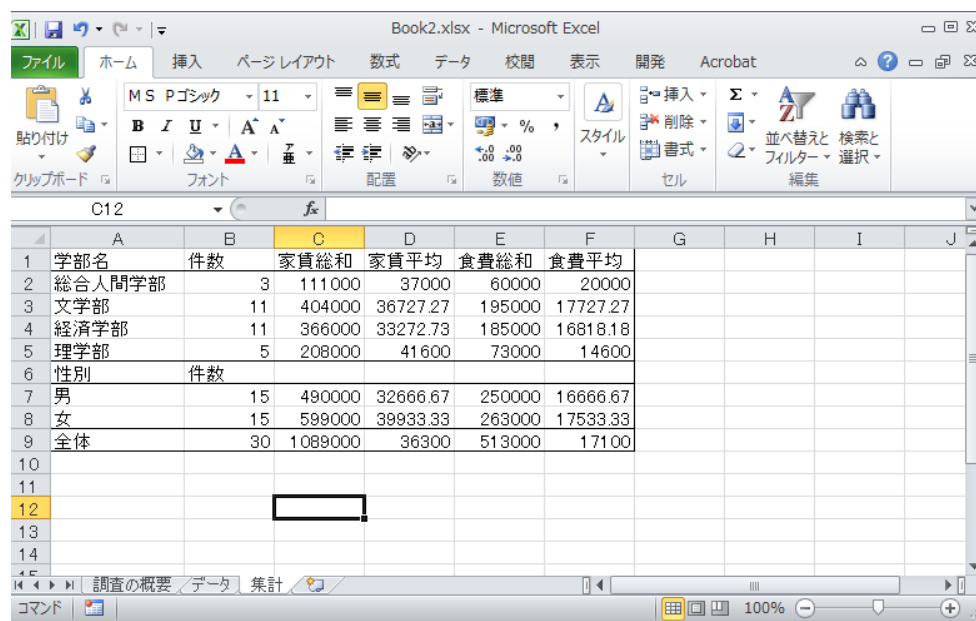
平均を求める関数には average, averegeif, averageifs などがあります。使い方は sum, sumif, sumifs 同様で、合計の代わりに平均が求まります。

演習 19. データの集計

演習 17 で作成したデータについて、sheet3 に“集計”と名前を設定して、以下を求めなさい。

- 全体のデータ件数、家賃、食費の総和と平均
- 学部別のデータ件数、家賃、食費の総和と平均
- 性別のデータ件数、家賃、食費の総和と平均

別のシートの範囲を選択するにはシート名に ! を付けて、例えば「データ!B2:B31」と書きます。



	A	B	C	D	E	F	G	H	I	J
	学部名	件数	家賃総和	家賃平均	食費総和	食費平均				
1	総合人間学部	3	111000	37000	60000	20000				
2	文学部	11	404000	36727.27	195000	17727.27				
3	経済学部	11	366000	33272.73	185000	16818.18				
4	理学部	5	208000	41600	73000	14600				
5	性別	件数								
6	男	15	490000	32666.67	250000	16666.67				
7	女	15	599000	39933.33	263000	17533.33				
8	全体	30	1089000	36300	513000	17100				
9										
10										
11										
12										
13										
14										

図 49 データの集計

また、検索条件は直接書く代わりに、検索条件を書いたセルを参照してもかまいません。上の例でセル C2 は条件として A2 を参照する形で以下のように書かれています。

6 表計算ソフトによるデータ処理（1）

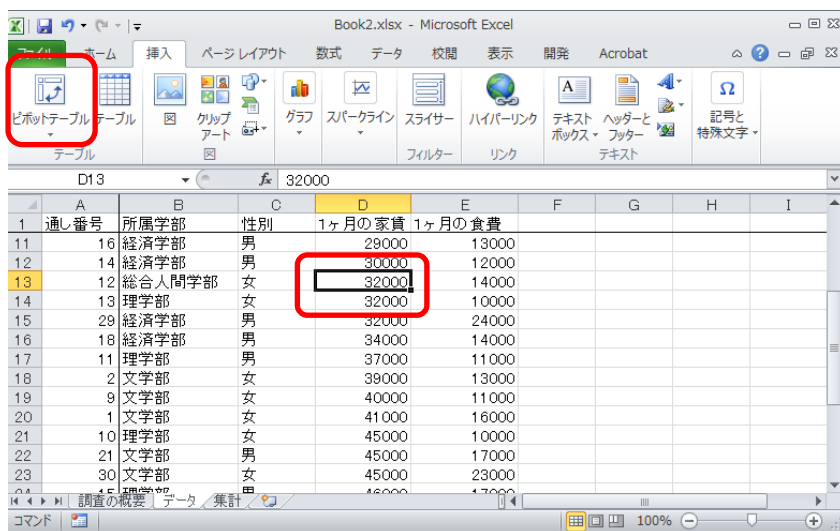
=SUMIF(データ!\$B\$2:\$B\$31,A2,データ!\$D\$2:\$D\$31)

また、データ範囲の設定は絶対参照で書かれていますが、これは下方にコピーした際に検索条件は A3, A4, A5 を参照しつつ、データ範囲は変えないためです。

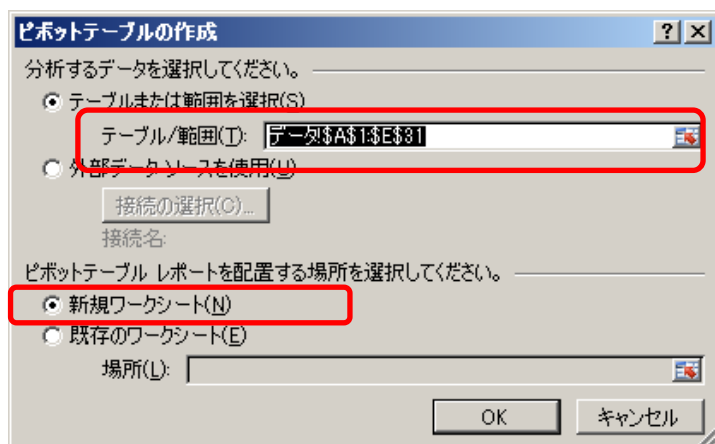
6.7 データのクロス集計

演習 17 で作成したデータに対して「学部別」と「性別」両方の条件ごとにデータ件数を求めたり、総和や平均を求めたりすることをクロス集計といいます。Excel でクロス集計するためには「ピボットテーブル」を使います。

- データのあるシート上でデータのあるセルを選びます。
- 挿入タブで表示されるリボンから「ピボットテーブル」を選びます。

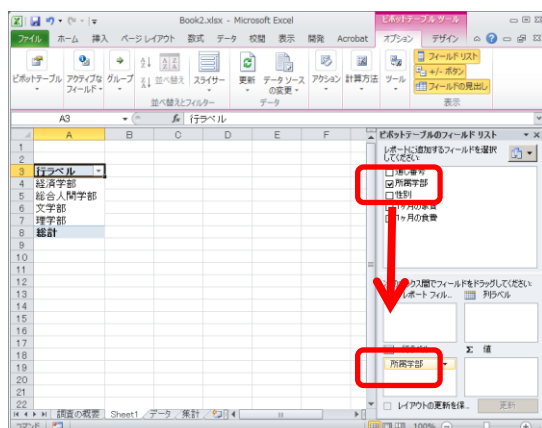


- データ範囲と新規ワークシート上に作成を確認して「OK」を押します。

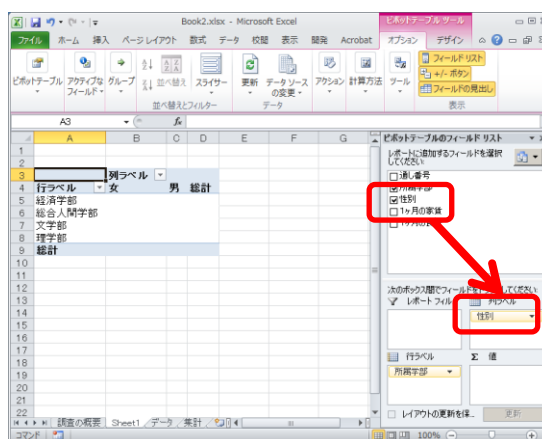


6 表計算ソフトによるデータ処理（1）

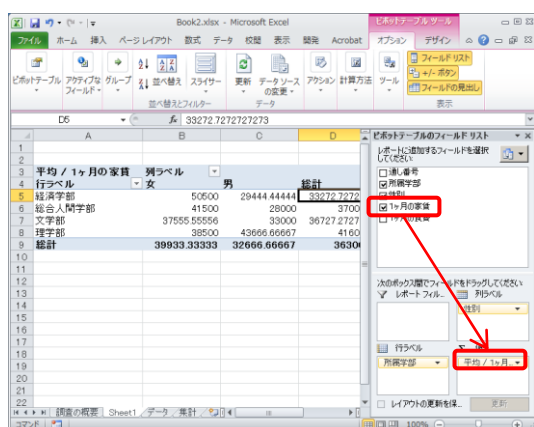
- ピボットテーブルのフィールドリストから所属学部を選び、行ラベルにドロップします。



- 性別を選び、列ラベルにドロップします。

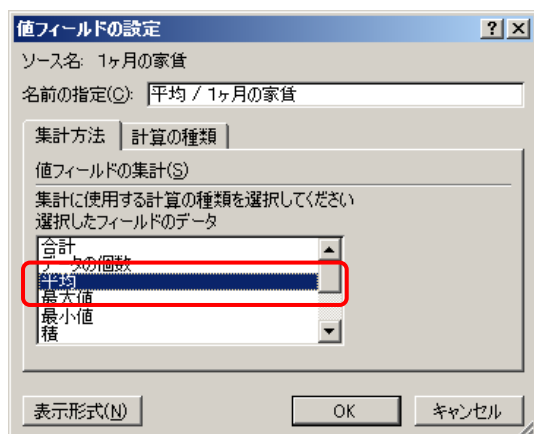


- 1ヶ月の家賃を選び、Σ値にドロップします



6 表計算ソフトによるデータ処理（1）

- Σ 値の合計／1ヶ月の家賃をクリックし、合計ではなく平均を選びます。



- 作られたシートのピボットテーブル以外のセルをクリックすればフィールドリストは消えます。ピボットテーブル上のセルを選べば再表示されます。

演習 20. クロス集計

演習 17 のデータに対してピボットテーブルを用いて学部別、男女別のデータ件数を求めなさい。

データの個数 / 1ヶ月の家賃	性別	合計
経済学部	男	9
経済学部	女	2
総合人間学部	男	1
総合人間学部	女	2
文学部	男	3
文学部	女	9
理学部	男	2
理学部	女	3
総計	男	15
総計	女	15
	合計	30

7. 表計算ソフトによるデータ処理（2）

7.1 学習の目的

- 表計算ソフトを用いて行うシミュレーションについて事例を用いて学びます。

7.2 シミュレーションとは

シミュレーション (simulation) とは考察の対象をモデルを用いて模擬することです。シミュレーションの対象としては物理的なシステムであったり，人が含まれる社会的なシステムであったりします。また，対象の変化の取扱いによって

- 定常的な状態にある様子を模擬する静的な状況のシミュレーション
- 時間とともに変化する様子を模擬する動的な状況のシミュレーション
- ランダムな要因を含むシミュレーション

という見方もできます。ここでは「**動的な状況のシミュレーション**」を取り上げて表計算ソフトでシミュレーションを実施することを学びます。

7.3 動的な状況のシミュレーションの構成要素

動的な状況を表すモデルの構成要素は以下のものです。

- 時間と共に変化する量（変数），これが計算の対象となります。
- 変数の変化を規定するルール。例えば1時刻前の値から次の時刻の値を得る関数（式）など。
- 変数の最初の時刻での値（初期値）
- パラメータ(parameter)。値の変化を規定するルールの中で参照される値をパラメータと言います。それ自体は時間とともに変化するものではありませんが，パラメータを変えたときに変数の値の変化の仕方などを考察します。

7.4 借入金の返済のシミュレーション

借入金は元本に利息をつけて返済しなければなりません。借入金と利息の一部ずつを返済して行くことで借入残高がどのように変化するかをシミュレーションしてみましよう。

7 表計算ソフトによるデータ処理 (2)

- 借入金は 1,000 万円とします。
- 利息は年に 3% とします。ただし端数は切り捨てるものとします。
- 1 年当たりの返済額は a 円とします。
- 借り入れは年初めに行うものとし、その年の終わりに利息がつくものとします。これから返済額を差し引いたものが、翌年の借入金残高となるものとします。すなわち

$$\text{翌年の借入金残高} = \text{int}(\text{前年の借入金} \times (1 + 0.03) - a)$$

ここで $\text{int}()$ は切り捨てて整数にする関数を意味します。

これを Excel で計算し、グラフ表示したものが図 50 です。上の図は計算結果を、下の図は式を表示したものです。計算結果を表示するか、式を表示するかの切り替えは Ctrl キーとシフトキーを押しながら 「`」 キーを押します。

- 式以外の箇所は文字列や値を直接入力します。E1, E2 セルにはそれぞれパラメータとして利率と 1 年当たりの返済額を設定します。
- 式はまず以下のように入力します

B3 セル $=\text{MAX}(\text{INT}(\text{B2}*(1+\text{\$E\$1})-\text{\$E\$2}),0)$

C2 セル $=\text{MIN}(\text{\$E\$2},\text{INT}(\text{B2}*(1+\text{\$E\$1})))$

それぞれ E1 セルの利率、E2 セルの毎年の返済額は絶対参照するために $\text{\$E\$1}$, $\text{\$E\$2}$ と書いています。MAX や MIN を用いているのは借入金残高が 0 以下にならないため、そして毎年の返済額が返済すべき額より大きくなったときに適切な値になるようにするためです。

- これらのセルをそれぞれ下方にコピーすることで表を完成させます。
- グラフの作成については次章を参考にしてください。

7 表計算ソフトによるデータ処理（2）

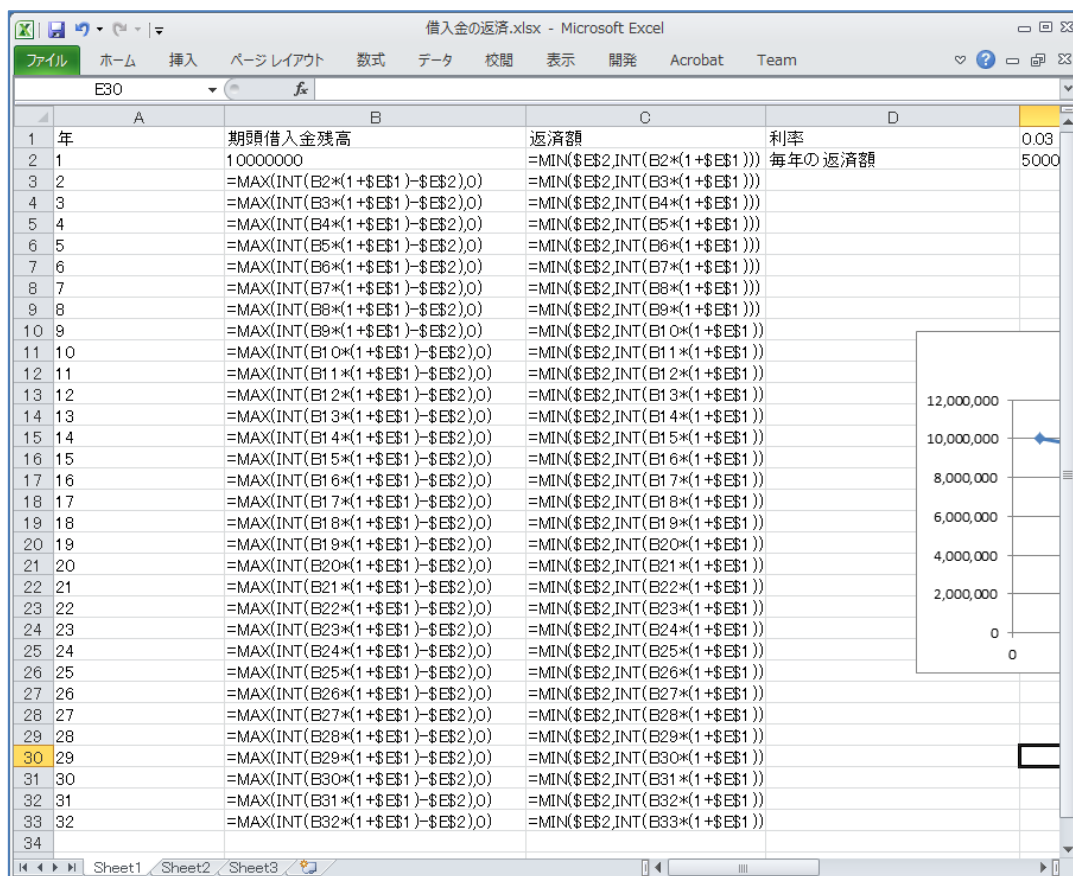
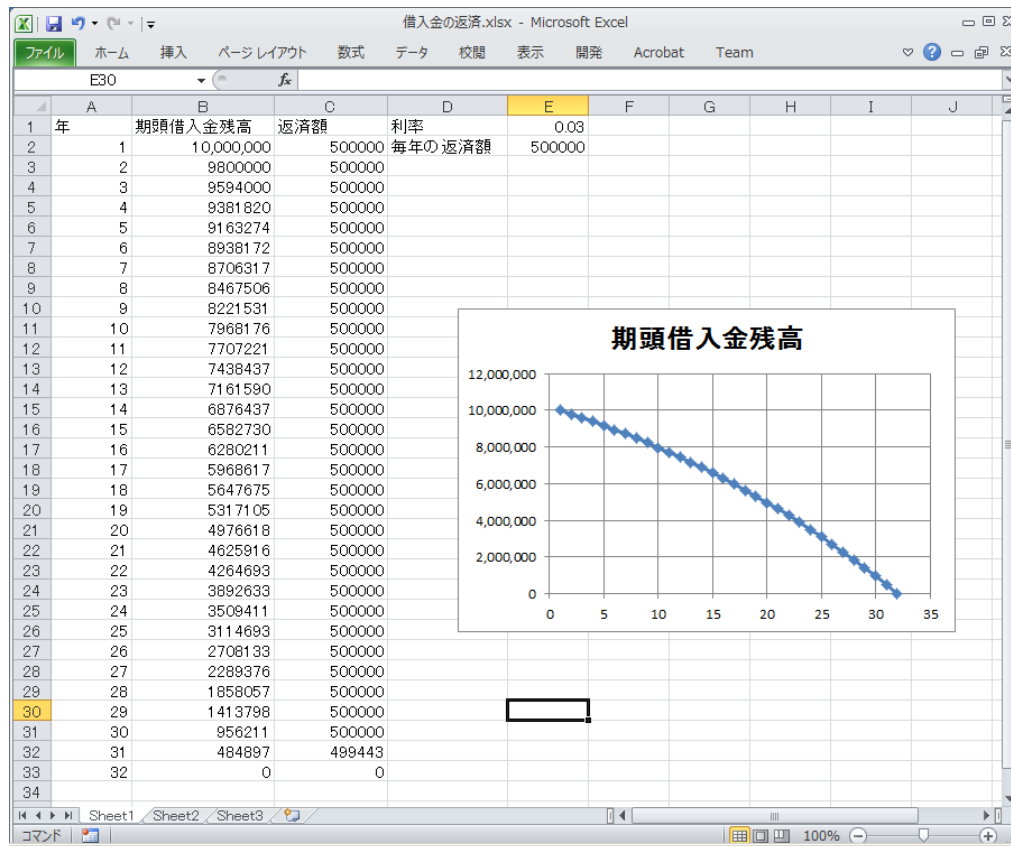


図 50 借入金返済のシミュレーション。下の図はセルの中の式を表示したもの

演習 21. 返済額を変えたシミュレーション

返済額を変えてシミュレーションを行い、借入金の残高の変化について考察しなさい。

7.5 バネ・ダンパモデルのシミュレーション

ここでは機械的なシステムのシミュレーションとしてバネ・ダンパモデルを取り上げます。

7.5.1 モデル

バネにおもりを吊るすことで、バネの伸びからおもりの重さを計測するのが「ばねばかり」の原理ですが、理想化されたバネでだとおもりを吊した場合、振動してしまいバネの張力とおもりの重力がつり合う長さで止まりません。このため動きを制動するための抵抗となる「ダンパ」が用いられます。このようなバネとダンパ、おもりから構成される仕掛けでのおもりの運動をシミュレーションすることを考えます。

おもりの質量を m 、バネ定数を k 、重力加速度を g 、ダンパの係数を c 、おもりの位置（つりあいの位置からのずれ）と速度を鉛直方向下向きを正にとってそれぞれ x 、 v とし、 x はバネの自然な長さを

原点にとるとするとおもりの運動方程式は次の式であたえられます。(1)式は速度の定義そのものです、(2)式は左辺の加速度（速度の微分）が右辺のように定数である重力、位置に比例するバネの復元力と速度に比例するダンパの抵抗力で与えられることを示したものです。

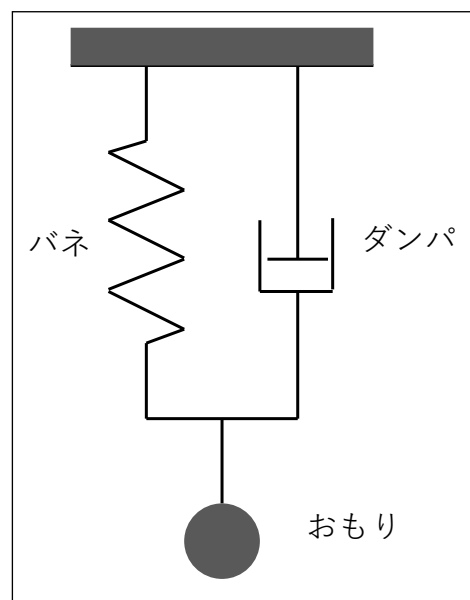


図 51 バネ・ダンパモデル

$$\frac{dx(t)}{dt} = v(t) \quad (1)$$

$$\frac{dv(t)}{dt} = g - \frac{k}{m}x(t) - \frac{c}{m}v(t) \quad (2)$$

またおもりの初期値は $x(0) = 0$, $v(0) = 0$ とします。

7.5.2 微分方程式のオイラー近似

微分方程式は連続時間の関数についての方程式ですから、そのままでは数値計算できません。ここでは短い時間間隔 Δt でおもりの位置や速度を計算して行くものとしします。すなわち $x_0 = x(0)$, $x_1 = x(\Delta t)$, $x_2 = x(2\Delta t)$, ..., $v_0 = v(0)$, $v_1 = v(\Delta t)$, $v_2 = v(2\Delta t)$, ... を考えます。そして、時刻 $n\Delta t$ から $(n+1)\Delta t$ の間は式(1), (2) で与えられる微係数は時刻 $n\Delta t$ の値で一定であると近似します。これにより、以下の差分方程式が得られ、数値計算ができます。

$$x_{n+1} = x_n + \Delta t v_n \quad (3)$$

$$v_{n+1} = v_n + \Delta t \left(g - \frac{k}{m} x_n - \frac{c}{m} v_n \right) \quad (4)$$

このような近似はオイラー近似と呼ばれ、微分方程式を数値計算する方法としては最も簡単なものです。ただし、時間間隔 Δt を大きく取ると精度が低下します。一方で小さく取ると数値計算に必要な計算量が大きくなります。本格的な数値計算にはより精度の高いさまざまな計算方法が開発されています。

7.5.3 表計算ソフトでのシミュレーション

式(3), (4)に基づいて Excel で行ったシミュレーションを図 52 バネ・ダンパモデルのシミュレーション図 52 に示します。パラメータは次の表のように設定しています。

表 7 バネ・ダンパモデルのパラメータのセルへの割り付け

パラメータ	セル
おもりの質量 m	E2
バネ定数 k	E3
重力加速度 g	E4
ダンパの係数 c	E5
時間の刻み Δt	E6

7 表計算ソフトによるデータ処理 (2)

これを用いて時刻、位置、速度の更新は例えば 3 列目 (A3, B3, C3) では以下のよう
に設定しています。借入金の返済シミュレーションと同様にパラメータはコピー
した際に参照位置がずれないように絶対参照を行っています。

$$A3 = A2 + \$E\$6$$

$$B3 = B2 + \$E\$6 * C2$$

$$C3 = C2 + \$E\$6 * (\$E\$4 - \$E\$3 * B2 / \$E\$2 - \$E\$5 * C2 / \$E\$2)$$

これらのセルを下方にコピーすることで時刻、位置、速度を更新するシートを作成
できます。

演習 22. ダンパの係数についての分析

ダンパの係数を 0 (この場合、単振動を生じます。) から徐々に大きくするとおも
りの運動はどのように変化をするかシミュレーションを用いて考察しなさい。

演習 23. ダンパの係数が 0 のときの吟味

ダンパの係数が 0 の場合は理論的には単振動を生じるはずですが、数値計算では振
動の振幅がどのように変化するのか観察しなさい。

7 表計算ソフトによるデータ処理（2）

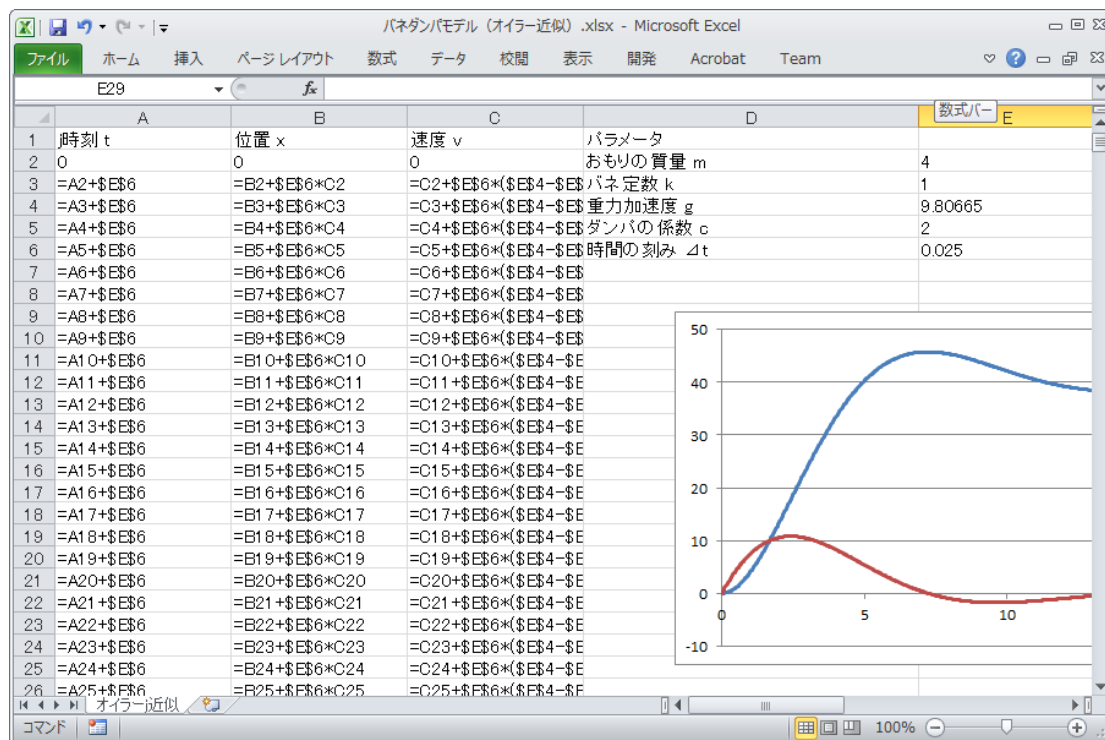


図 52 バネ・ダンパモデルのシミュレーション

7.5.4 静的な状況とその扱い

静的な状況をバネ・ダンパモデルで考えてみましょう。時間的に変化しないのですから、式(1)、(2)で表される微係数が 0 になると考えます。すなわち

$$0 = v(t) \quad (5)$$

$$0 = g - \frac{k}{m}x(t) \quad (6)$$

これはバネの張力とおもりに働く重力がつりあっている状態にほかなりません。この例では容易に解くことができますが、より複雑な微分方程式で表された対象が静止している状態は微係数を 0 とおいて得られる連立代数方程式を解くことが求められます。

8. 表計算ソフトによるデータ処理（3）

8.1 学習の目的

- データをグラフとして表現することの意義と注意事項を考えます。
- Excel でグラフを作成する際の用語を学びます。
- Excel で散布図を描く方法を学びます。
- Excel で折れ線グラフや棒グラフを描く方法を学びます。
- Excel でヒストグラムを描く方法を学びます。

8.2 データをグラフで表示することの意味

演習 24. レポートなどではしばしばデータをグラフで表示します。なぜグラフで表示するのでしょうか？

演習 25. データをグラフで表示することは知的生産活動の中でどのようなことに役立つでしょうか。

8.3 データをグラフで表現する時の注意事項

次のグラフは演習 17 で示したデータについて横軸に 1 ヶ月の家賃を、縦軸に 1 ヶ月の食費をとってグラフに示したものです。レポートなどを行う際にこのようなグラフで示された場合に読み手に情報は十分に伝わるでしょうか。

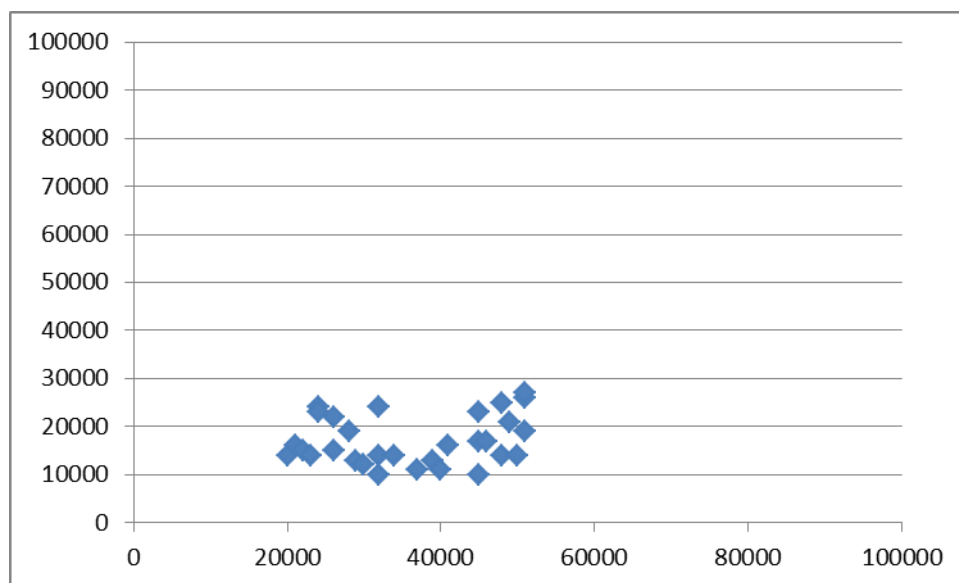


図 53 あまり適切でないプロット

演習 26. この問題点を挙げてみてください。

8.4 グラフを Excel で作成する際の用語

Excel でグラフを描く際にはグラフを構成する要素の名称を確認しておくとい分りやすいです。図 54 は言葉が示す対象を分りやすくするため背景を着色しています。

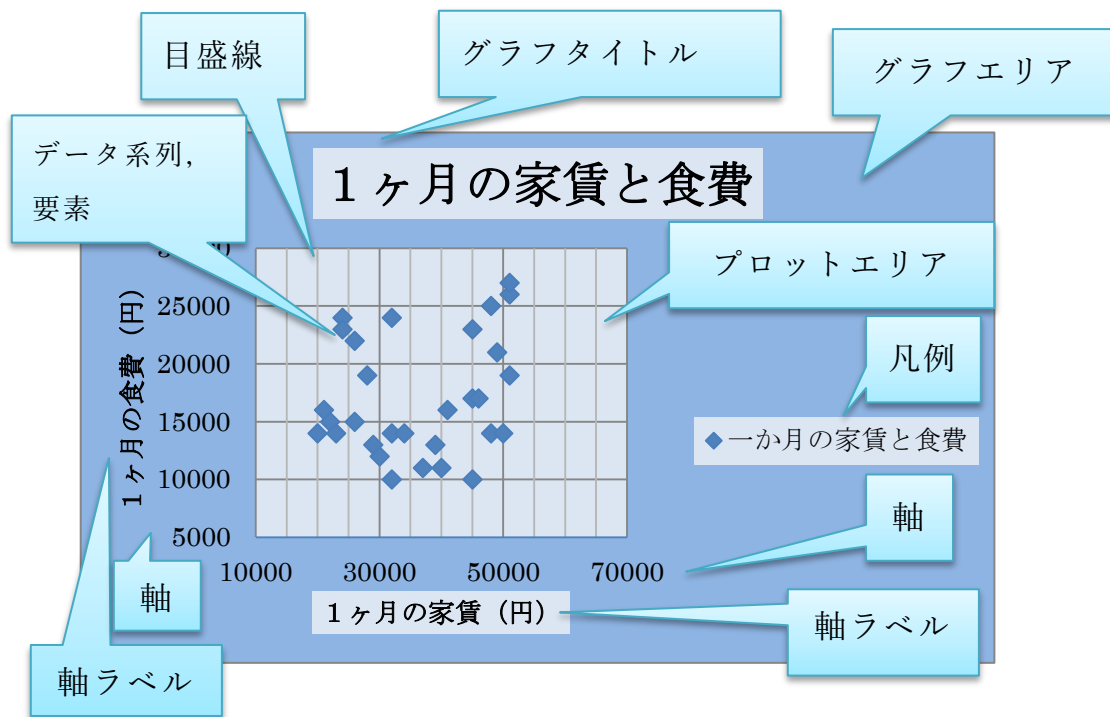


図 54 Excel でのグラフ各部の名称

8.5 散布図を Excel で描く

8.5.1 散布図の用途

先の例に示したグラフは**横軸と縦軸の値を与えてデータをプロットする**もので Excel では「散布図」という形式です。以下のような使い方があるでしょう。

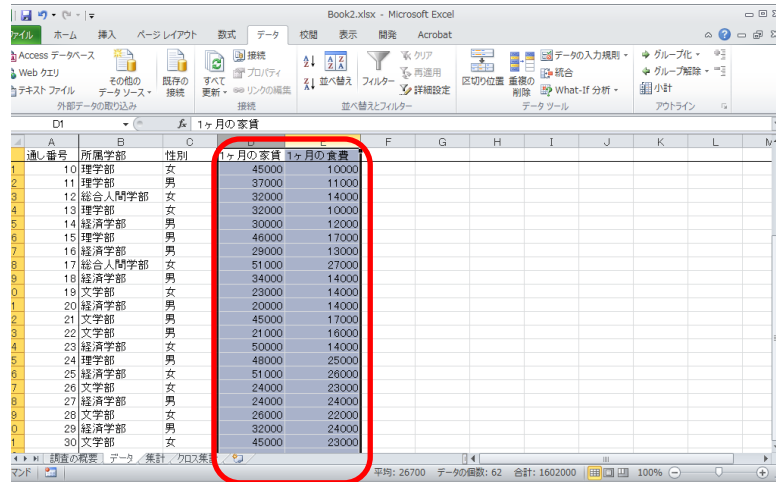
- アンケート調査などのデータで2つの属性の関係を見るとき使う。
- 実験などで横軸の値を設定して、縦軸の値を計測し、両者の関係をグラフに表す。通常、折れ線グラフなどで描きますが、横軸の値を自由に与えてプロットする際には散布図を選びます。

8.5.2 散布図の作成手順

前回の演習 17 のデータを例に散布図を描く手順を以下に示します。

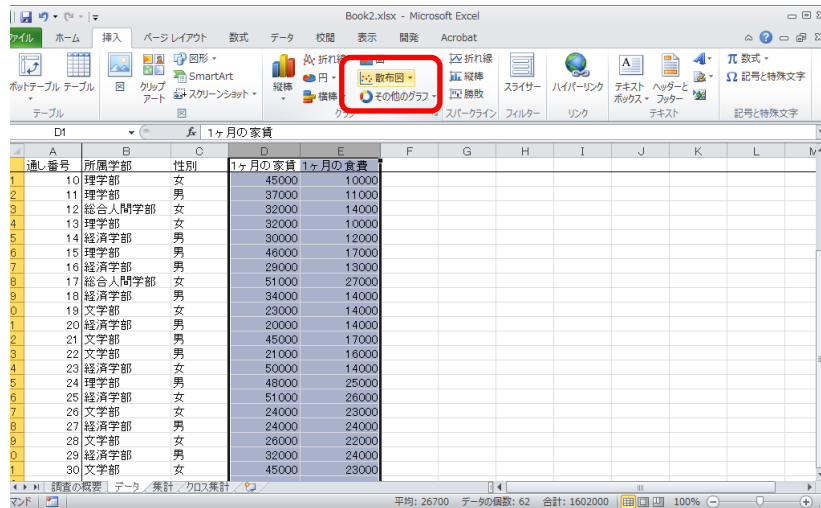
8 表計算ソフトによるデータ処理（3）

1. 散布図を描くデータ列を2つ（例では「1ヶ月の家賃」と「1ヶ月の食費」）を選びます。



通し番号	所属学部	性別	1ヶ月の家賃	1ヶ月の食費
10	理学部	女	45000	10000
11	理学部	男	37000	11000
12	総合人間学部	女	32000	14000
13	理学部	女	32000	10000
14	経済学部	男	30000	12000
15	理学部	男	46000	17000
16	経済学部	男	28000	13000
17	総合人間学部	女	51000	27000
18	経済学部	男	34000	14000
19	文学部	女	23000	14000
20	経済学部	男	20000	14000
21	文学部	男	45000	17000
22	文学部	男	21000	16000
23	経済学部	女	50000	14000
24	理学部	男	48000	25000
25	経済学部	女	51000	26000
26	文学部	女	24000	23000
27	経済学部	男	24000	24000
28	文学部	女	26000	22000
29	経済学部	男	32000	24000
30	文学部	女	45000	23000

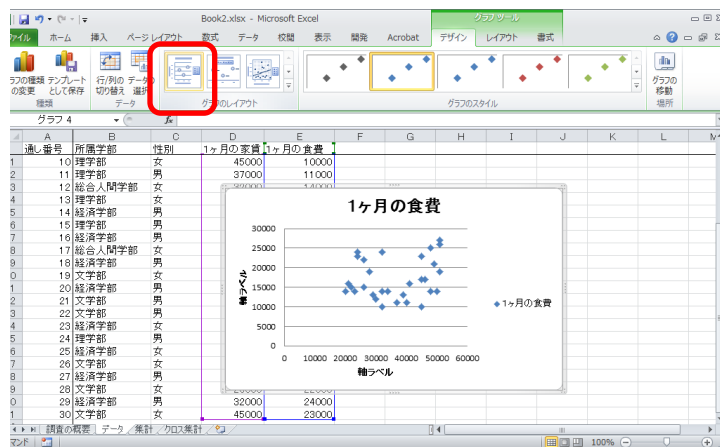
2. 「挿入」タブを選び、グラフのリボンの「散布図」を選び、プロットする形式（この場合はマーカーだけ）を選びます。



通し番号	所属学部	性別	1ヶ月の家賃	1ヶ月の食費
10	理学部	女	45000	10000
11	理学部	男	37000	11000
12	総合人間学部	女	32000	14000
13	理学部	女	32000	10000
14	経済学部	男	30000	12000
15	理学部	男	46000	17000
16	経済学部	男	28000	13000
17	総合人間学部	女	51000	27000
18	経済学部	男	34000	14000
19	文学部	女	23000	14000
20	経済学部	男	20000	14000
21	文学部	男	45000	17000
22	文学部	男	21000	16000
23	経済学部	女	50000	14000
24	理学部	男	48000	25000
25	経済学部	女	51000	26000
26	文学部	女	24000	23000
27	経済学部	男	24000	24000
28	文学部	女	26000	22000
29	経済学部	男	32000	24000
30	文学部	女	45000	23000

3. 描かれたグラフを選んで、リボンから適当なレイアウトを選びます。

8 表計算ソフトによるデータ処理（3）



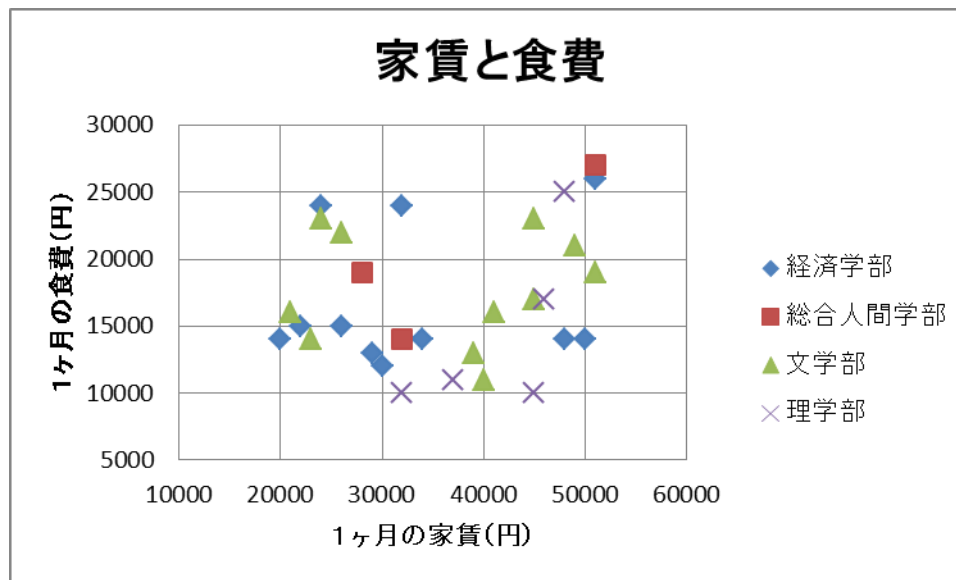
4. 横軸を選んで右クリックし、「目盛線を追加」を選びます。



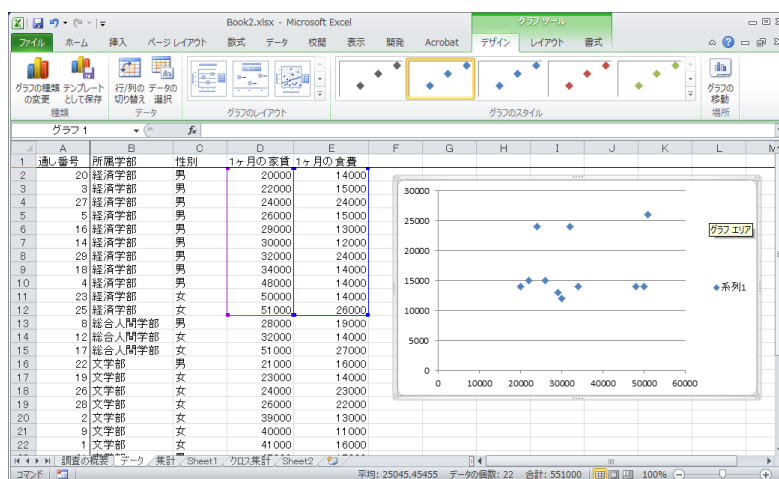
5. グラフタイトル，軸ラベル，凡例などを適宜修正します。

8.5.3 複数系列の散布図

以下のような学部別の散布図を作成してください。

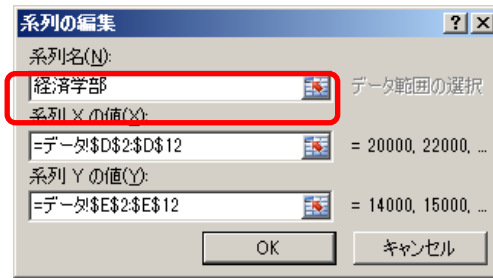
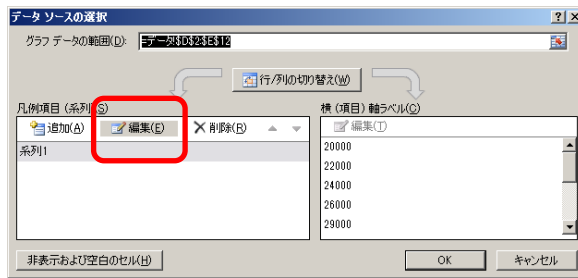


1. まず、データを学部で並べ替えます。
2. 経済学部について、家賃と食費のデータを選択して散布図を描きます。

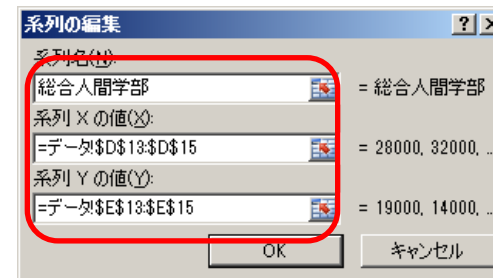
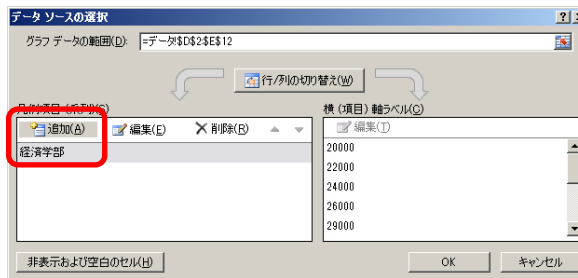


3. プロットされたデータを選んで右クリックし、データの選択を選びます。系列1を編集して系列名を設定します。

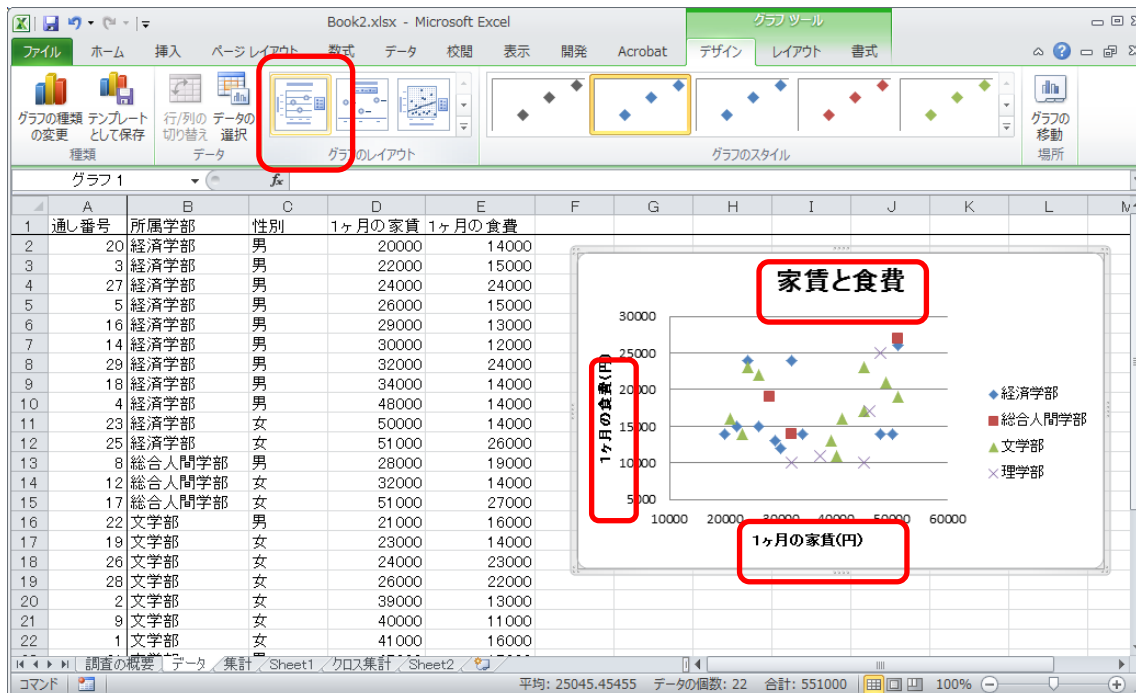
8 表計算ソフトによるデータ処理（3）



4. データの追加を選び、総合人間学部の家賃と食費のデータを設定します。



5. 同様に文学部と理学部についても作業します。
6. レイアウトを選びグラフタイトルや軸ラベル、目盛線などを設定します。



8.6 棒グラフや折れ線グラフを Excel で描く

Excel で棒グラフや折れ線グラフを描くことは散布図より分かりやすいと思います。次の例はデータを家賃の昇順で並べ替えて折れ線グラフでプロットしたものです。

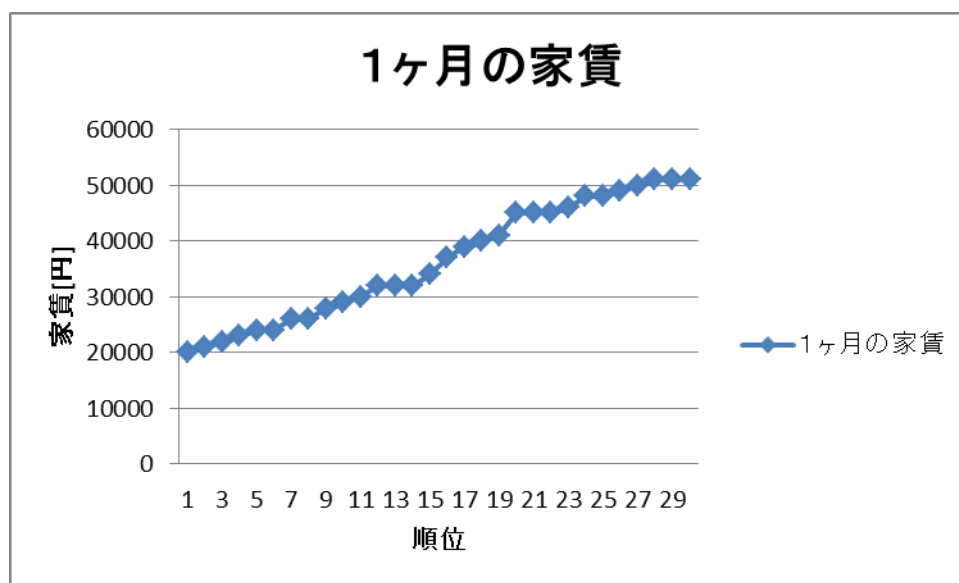


図 55 折れ線グラフの例

また、次のグラフはピボットテーブルを使ってクロス集計した家賃の平均を縦棒グラフで表示したものです。ピボットテーブル内で適当にデータを選択し、「挿入」→「縦棒」グラフ→「集合縦棒」と選ぶと自動的にピボットテーブルの内容をグラフ表示できます。

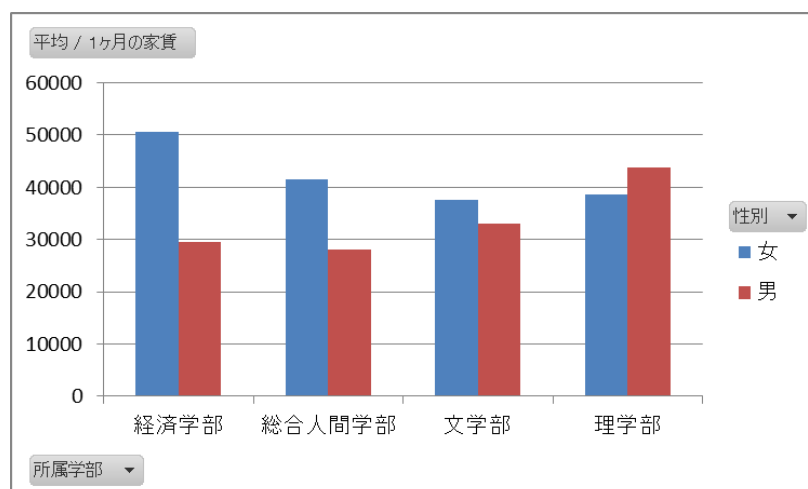


図 56 棒グラフの例

8.7 ヒストグラムを描く

以下のような家賃の分布をヒストグラムとして描いてみましょう。

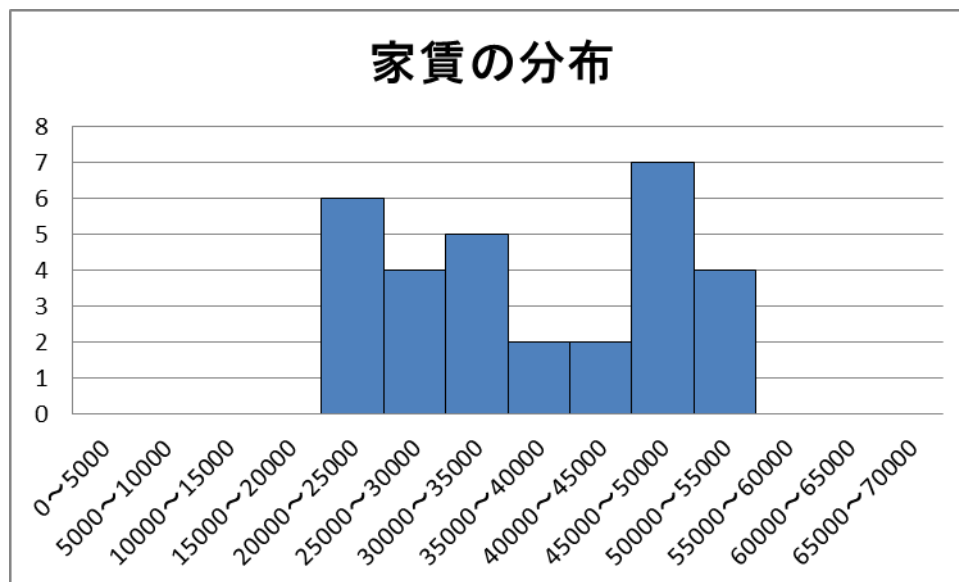


図 57 ヒストグラム

8.7.1 ヒストグラムとは

横軸に階級，縦軸に度数（該当する件数）を取って描いたグラフです。階級を規定するデータの所属する区間のことをビン(bin)と言います。ビンの幅が狭いと細かな表現が得られますが，各ビンに含まれるデータが少なくなり分布を掴むことは難しくなります。他方でビン幅が大きいと，分布の細かな状況が分からなくなります。

8.7.2 データの作成

家賃のデータを参照して，指定された範囲に入るデータの件数を数えます。

8 表計算ソフトによるデータ処理（3）

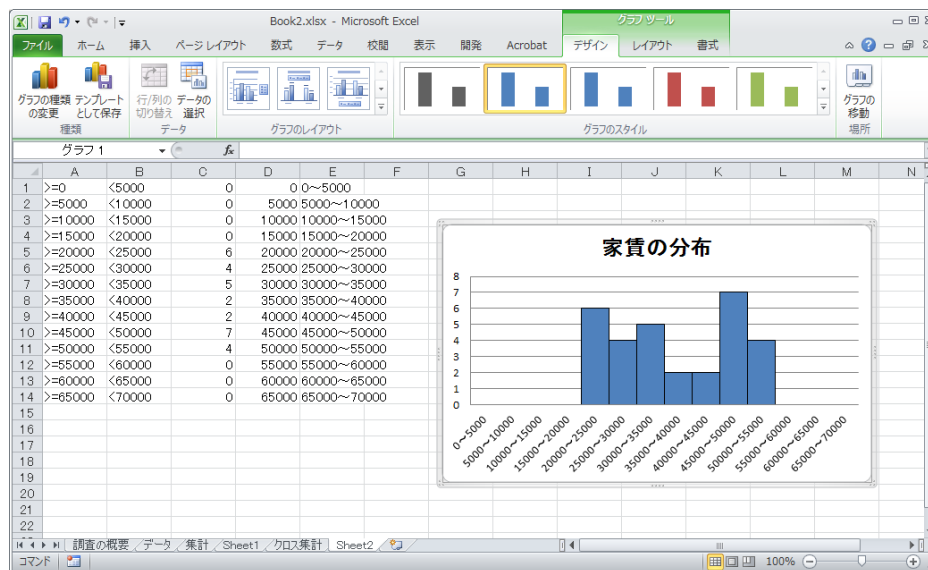


図 58

この例では新しいシート (Sheet2) をヒストグラムのデータ作成用に用いています。

- A 列に範囲の下限を与える式 (A1 だと “>=0”) を,
- B 列に範囲の上限を与える式 (B1 だと “<5000”) を書き,
- C 列については例えば C 1 セルの内容を

=COUNTIFS(データ!\$D\$2:\$D\$31,Sheet2!A1,データ!\$D\$2:\$D\$31,Sheet2!B1)

として、条件に合うデータの個数を数えています。

- D 列には範囲の下限を数値だけ作成し,
- E 列は軸ラベル用の文字列を

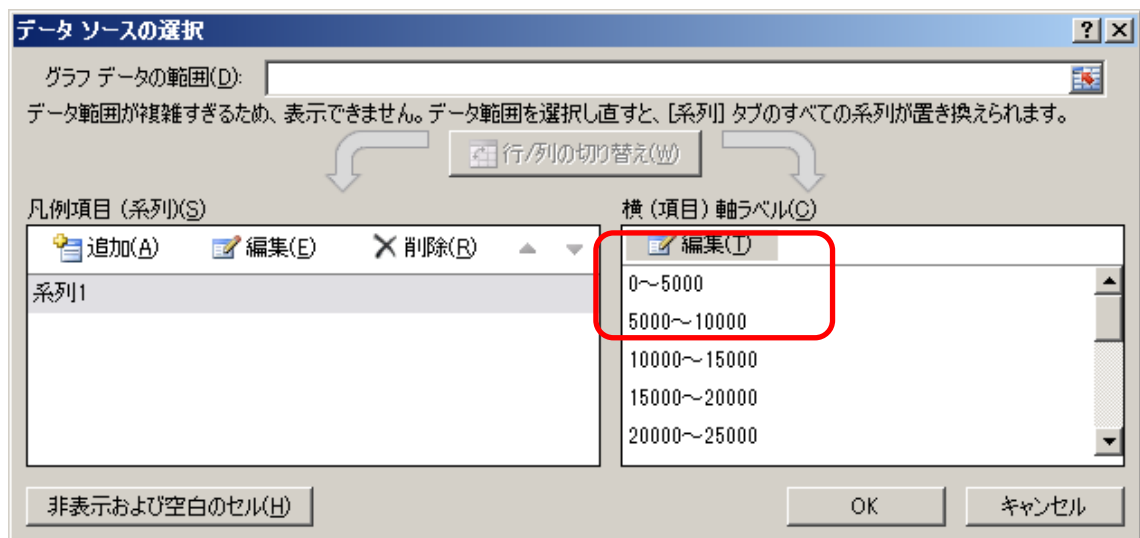
=D1&"~"&(D1+5000)

(E 1 セルの内容の例) として作成しています。

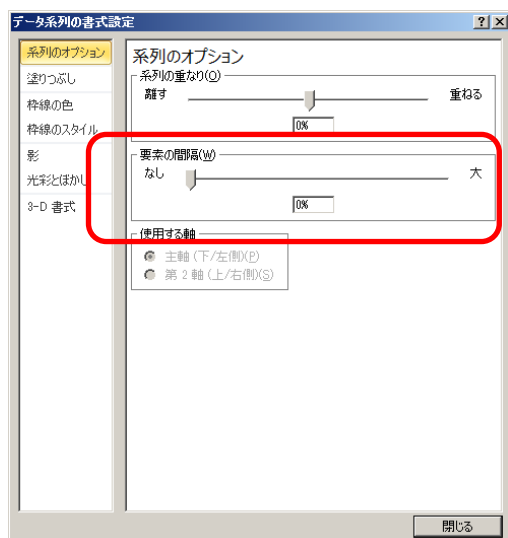
8.7.3 グラフのプロット

1. C 列の計算結果を選んで縦棒グラフでプロットします。
2. データ系列を選んで右クリックし、「データ選択」を選び、横（項目）軸ラベルの編集を選んで、軸ラベルとして E 列の値を指定します。

8 表計算ソフトによるデータ処理（3）



3. ヒストグラムは縦棒の間を空けません。データ系列を選んで右クリックし、「データ系列の書式設定」を選び、系列のオプションで「要素の間隔」を「なし」にします。また枠線の色を適宜設定します。



演習 27. 家賃の分布のヒストグラムを作成しなさい。

9. 構造化された文書の作成

9.1 学習の目的

- 論文やレポートの文章がどのように構造化されているかを学びます。
- Word での段落の扱いと章立ての方法を学びます。
- Word での参考文献の記載やその引用の方法を学びます。
- Word での文章の校閲の方法を学びます。

9.2 論文やレポートの構造

論文やレポートは論理的に物事を表すために以下のような構造を持たせて書かれています。

- **題目，著者，日付**
- **要約**
- **章立てと見出し**，章，節などの形で章立てが行われ，しばしば見出しに番号が振られています。番号は前後の章を他の箇所で参照する際に使います。著者が参照しなくても，その文章を引用したり，批評したりする他の著者によって使われることもあります。章や節の見出しのフォーマット（フォントや大きさ，行の揃え方）は文章内で一貫した表記法がとられます。本文以外に謝辞や付録が付けられることがあります。
- **パラグラフ**，学術的な文章ではまとまった内容の文章のかたまりを「**パラグラフ，paragraph**」として表記します。パラグラフの冒頭を下げたり（**字下げ，インデント，indent**），パラグラフの間の行間を長めにとったりすることで，視覚的にパラグラフであることを分かりやすくします。

- **箇条書き**，並列する内容，順序のある内容（項目，アイテム，item）を分かりやすくするために箇条書きが用いられます。項目の順序を明示的に示したり，項目の参照を容易にしたりするために通し番号が付されることがあります。
- **注**，本文の内容を補足するものではあるが，本文の流れを妨げたくない場合に脚注や文末注が用いられます。
- **参考文献とその引用**，論文，レポートではその情報源を明示することで，議論している内容の論拠を示します。通常，文章末に参考文献のリストを付け，文章内でそれを引用する形で書きます。文献や引用は読者がその情報源に的確に辿りつけるだけの情報（書誌情報）が含まれることが求められ，標準的な記法が提唱されていたり，学術雑誌等で定められた書式に従う必要があったりします。
- **図や表の挿入とその参照**，本文中に図や表を挿入し，**図番号と説明（キャプション，caption）**，**表番号と標題（タイトル，title）**が付されます。図番号と説明は図の下に，表番号と標題は表の上に入れられることが多いです。本文中では図や表を図番号，表番号で参照します。
- **数式**，数式を本文中に挿入したり，独立した行として挿入したりします。

演習 28. 文献例の構造を探る

指示された文献についてどのような構造と表現上の書式が使われているかに留意して読んでください。

9.3 人がすること，コンピュータに任せること

メールなどで行事や会議の開催日を通知するときに日付に曜日をつけて書きますが，しばしば曜日が誤って付けられ，後で訂正されることがあります。なぜ，曜日を誤って書いてしまうのでしょうか，どうすれば誤りはなくせるのでしょうか？誤る理由は，日付と曜日のどちらもを人間が直接キーボードで入力しているからだと推察されます。他方で，日付（年月日）が決まれば曜日は自動的に決まるもののはずです。もしそうなら，人

は年月日のみを入力し、曜日はコンピュータに補ってもらう方がいいのではないのでしょうか。

レポートなどを書いていると、章や節に番号を振ったり、振られた番号を用いて文中で他の章や節を番号で指示したりします。番号を振ることは単純な作業ですが、文章の執筆中は章を追加したり、削除したりします。その度に人手で番号を振り直し、章番号を引用している箇所を修正したりしなければなりません。短い文章ではすべて手作業でやることも考えられますが、少し量が多くなればこの作業は手間がかかり、修正漏れも発生するでしょう。それならば、**章に番号を振ったり、引用している章番号を常に正しい値に維持したりすることはコンピュータに任せてしまう**、という考えに至ります。

またレポートでは見出しなどを付けますが、番号を付けるだけでなく、目立たせるために本文に比べ文字の書体を変えたり、大きめにしたりします。これについても、1) **文章には「見出し」としての指定だけを行って**、2) **「見出し」として指定された個所の表現方法は別途、設定します**。これにより、見出しの表現を統一的に調整することをコンピュータに任せます。

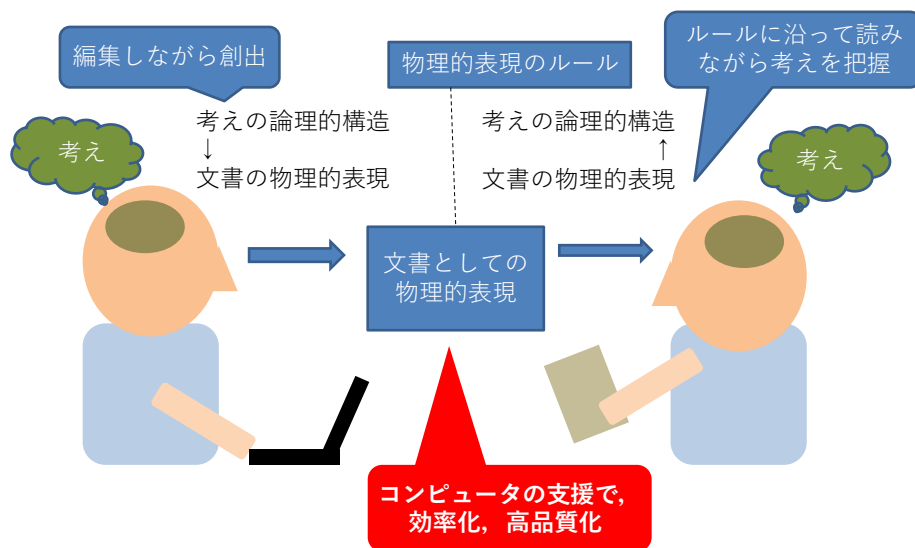


図 59 「考え」の文章化と文章からの「考え」の獲得

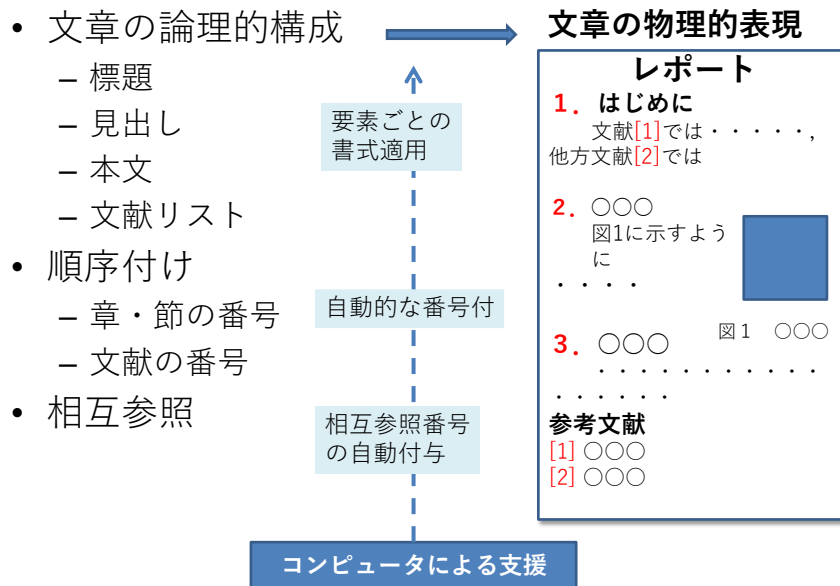


図 60 文章の論理的構成から物理的表現への変換

9.4 スタイルとテンプレート

Word には標題、見出し、本文、参考文献リストといった文章を構成する個々の要素に対して、効果的、効率的に書式を自動的に付与する仕掛けがあり「**スタイル**」と呼ばれます。そして、一連のスタイルを収めたファイルを「**テンプレート**」と呼びます。例えば新規文書を作成するときには「空白の文書」を選ぶことが多いと思いますが、これは「**空白の文書**」という**テンプレート**を適用して、文書ファイルを作成することを意味します。

9.4.1 読みやすさを重視したスタイルの必要性

学術や業務などの目的で書かれるレポートなどの文章は**読みやすさに配慮することが重要です**。Word では書体（フォント）をさまざまに選択することが可能ですが、

- 通常の文章は明朝体で、
- 見出しなど強調する箇所はゴシック体で書くことをおすすめします。

英文については明朝体やゴシック体をそのまま使うのではなく英文用

9 構造化された文書の作成

のフォントを使うと読みやすくなります。和文と英文が混在する場合は似た書体を組み合わせて使います。例えば、以下のような組み合わせになります。(和文フォントのうち、游明朝、游ゴシックは Windows7 では標準では導入されていません。英文フォントについては文献[8]で推奨されているものを挙げました。

表 8 レポートで使うフォント

和文	和文フォント	和文フォント	英文フォント
明朝体	游明朝	MS 明朝	Times New Roman
ゴシック体	游ゴシック	MS ゴシック	Segoe UI

文字の大きさはレポートでは 10.5 ～ 12 ポイント程度で選びます。行間は文字の大きさの 0.5 ～ 1 倍に選ぶようにします[8]。

9.4.2 「空白の文書」のスタイルの指定

「空白の文書」をテンプレートに新しい文書を作成した場合、「空白の文書」で定義されているスタイルが利用できます。よく使うスタイルはホームで表示されるリボンの「スタイル」のなかのクイックスタイルギャラリーに表示されていますが、利用できるスタイルの一覧を表示するには、リボンのスタイルの右下の矢印をクリックしてください。

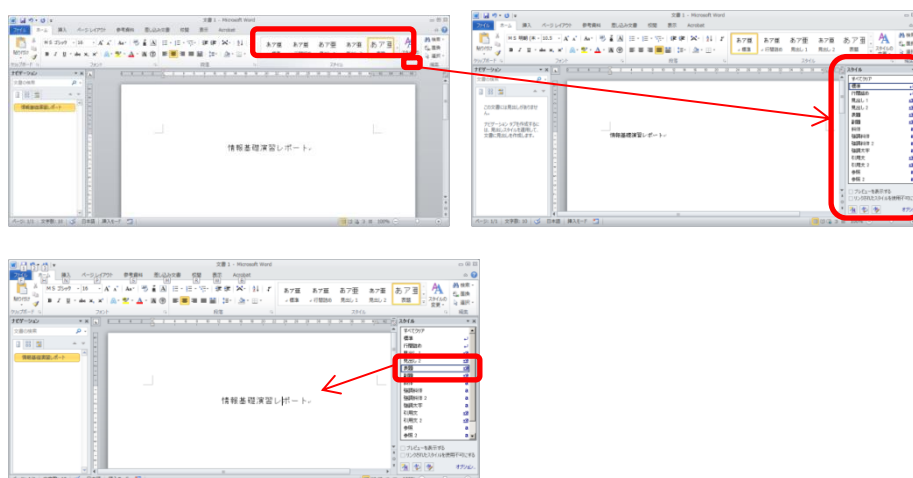


図 61 スタイルの適用

9.4.3 レポート用テンプレート

「空白の文書」で設定されているスタイルは章見出しの番号付けなどは必ずしも大学でのレポートに適したものではありません。レポートの作成を支援するために

レポート論文用表紙なし 20160402.dotx

というテンプレートを用意しました。このテンプレートの使い方については、資料「レポート論文用表紙なしテンプレートの使い方」を参照してください。

9.5 パラグラフの書き方

Word では 1 つのパラグラフの中では改行せずに文をつづけます。スタイルを「標準」あるいは「本文」とし、字下げは文書の上にあるルーラーで調整します。字下げの整形のために文頭に空白を挿入することは望ましくありません。なぜなら、先頭に文を追加するなどして、パラグラフの構成を変更したりする場合、空白は段落中に残ってしまい、これを消すことを見落とす可能性があるからです。

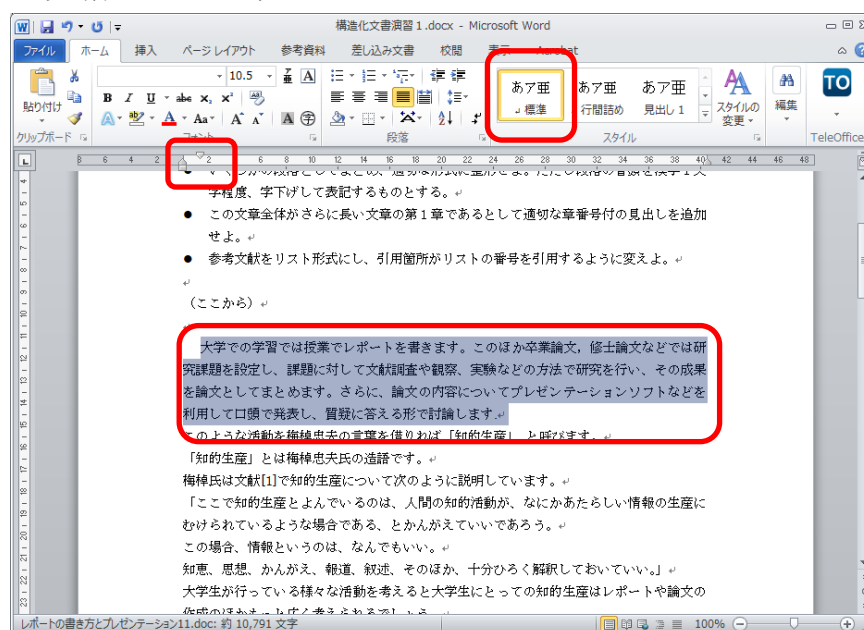


図 62 パラグラフ先頭の字下げ

9.6 章立ての見出しの書き方

章立ての見出しは以下の手順で書きます。

1. 見出しを書きます。
2. 見出しのスタイルをそのレベルに合わせて「見出し 1」、「見出し 2」など選びます。
3. 段落リボンからアウトラインボタンを選び、見出し番号の形式を選びます。**アウトラインボタンのうち「見出し」という言葉が含まれるものを選びます。**そうでないものは箇条書き用のものです。

なお、Word の「空白の文書」として設定されている見出しの番号形式にはレポートでしばしば用いられる 1. や 2.3 といったものが設定されていません。これについては、新しいアウトラインボタンを押して現れるメニューから「新しいアウトラインの定義」を用いて調整します。

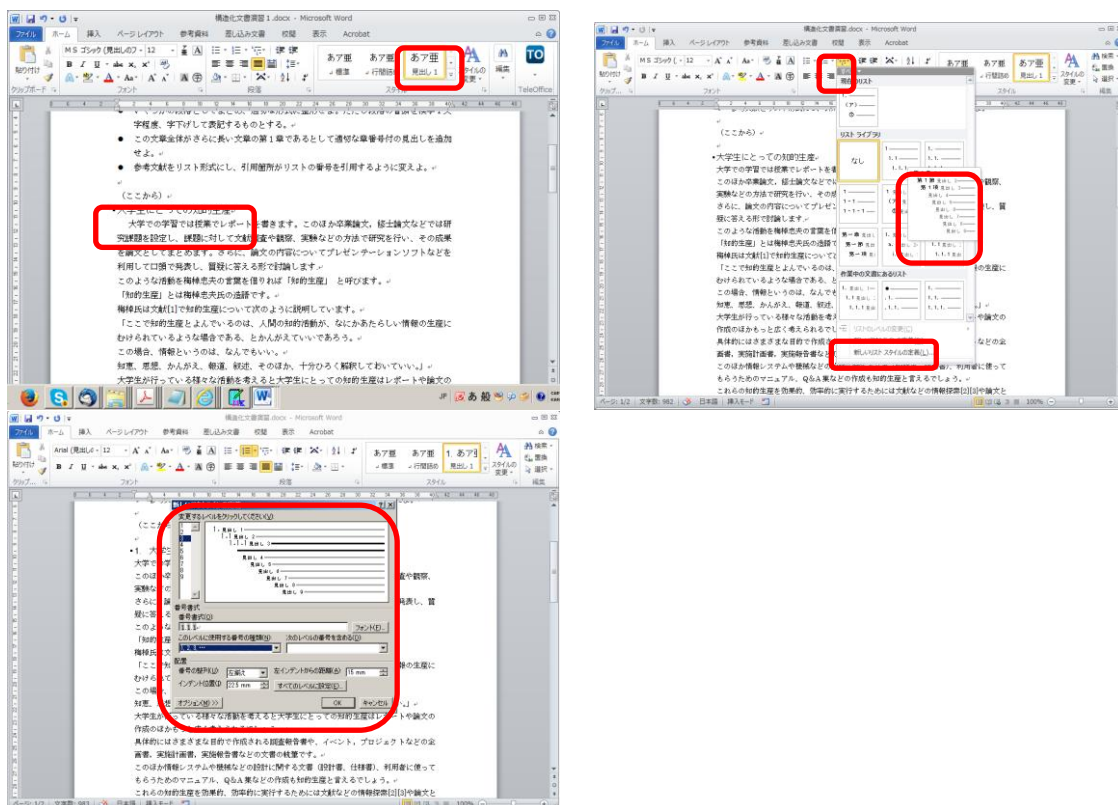


図 63 見出し用スタイルの調整

また、「参考文献」は見出しに番号をつけないことが多いですが、この

9 構造化された文書の作成

場合は、参考文献という見出しに対して、アウトラインボタンで「なし」を選んでください。

演習 29. パラグラフと見出しの設定

演習課題用文章に対して、指示に従ってパラグラフと見出しをつけなさい。

9.7 アウトライン表示の活用

9.7.1 アウトライニング

学術的な文章のように、多数の資料などを参照しながら論理立った文章を書くことは決して容易ではありません。文章を執筆するための素材から、文章の構成の骨格（アウトライン）を考え、これをパラグラフとしてまとめていく形で文章を作成することは、文章作成のためのよい方法です。

9.7.2 アウトライン表示の利用

Word ではアウトライン表示を使うことでこの作業の効率を高めることができます。「表示」メニューで現れるリボンの「文書の表示」から、「アウトライン」を選びます。

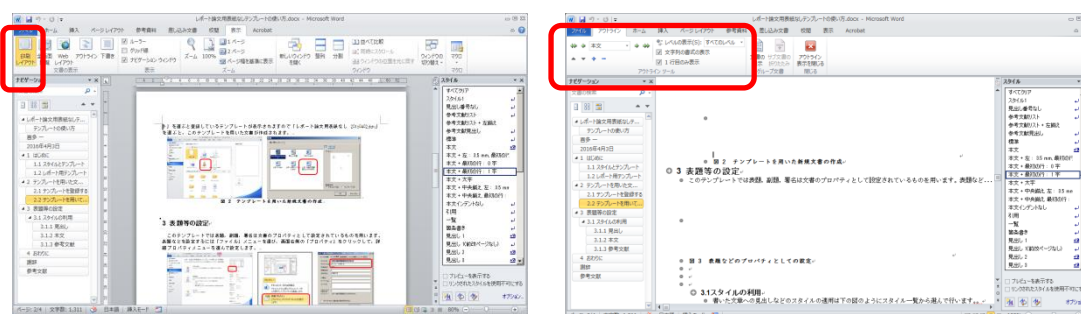


図 64 アウトライン表示

- アウトラインツールの「+」「-」で章や節を展開したり折りたたんだりできます。

- ▽や△で選択した章や節全体を上下に移動することができます。

9.8 参考文献の記載方法とその引用

9.8.1 参考文献リストの書き方

ここでは参考文献は単純に番号付けされた箇条書きで実現しています。また「参考文献」という見出しを「見出し1」のスタイルで設け、特に番号は付さない形にしています。

参考にする資料には書籍、論文、Web サイトなど多様なものがあります。参考にしたものは必ず参考文献として掲載することと、その記載（書誌情報）については読者が適切にその内容を確認できるための情報が得られることが基本です。参考文献の記載方法については「科学技術情報流通技術基準」[11]を参照するとよいでしょう。

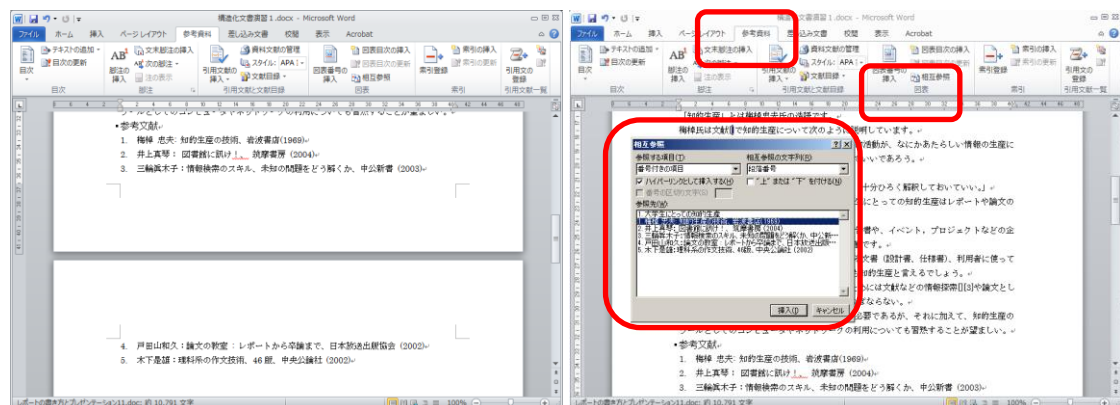


図 65 参考文献リストの記載と参照

9.8.2 引用の書き方

本文中などで文献を引用する場合、文献リストの番号を挿入する必要があります。先のように番号付の箇条書きにしてあれば、「参考資料」というメニューのリボンにある「相互参照」で文献の番号を挿入できます。

9.8.3 引用などの更新

参考文献のリストに追加や削除があった時は該当箇所（文章全体なら「すべてを選択」）を選んで右ボタンをクリックし、「フィールドの更新」を選べば文献番号が付け直されます。

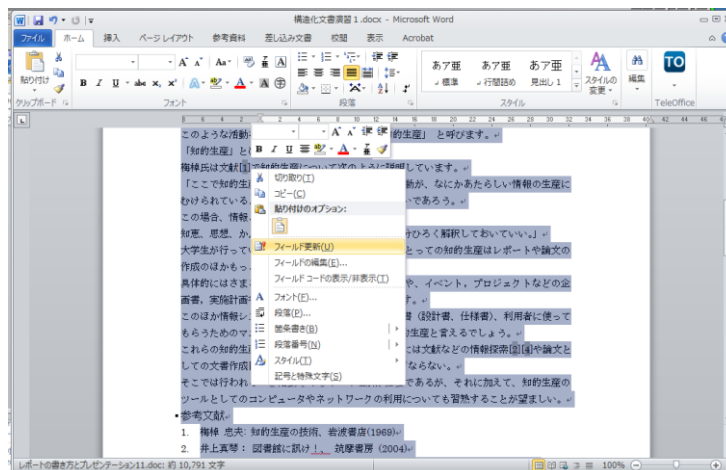


図 66 フィールドの更新による相互参照の更新

演習 30. 参考文献リストの作成と相互参照

演習課題用文章について、参考文献を箇条書き形式で作成し、引用箇所について相互参照で書き直しなさい。またフィールドの更新について操作を確認しなさい。

9.9 文章の校閲方法

9.9.1 文章の校閲とその他の Word の機能

文章は何度か書き直して仕上げて行きますが、その際のどこを修正したのかが分かることが望めます。とりわけ、グループで文章を作成している場合や指導教員から指導を受ける場合など修正箇所を分かりやすくすることはコミュニケーションの質を高める上でも重要です。

Word では「校閲」というメニューに変更箇所の記録を取れる「変更履歴の記録」という機能があります。また、文章の特定の箇所に対して注釈をつける「コメントの挿入」という機能があります。

演習 31. 変更履歴の利用とコメントの付与

演習課題用文章について変更履歴を記録した上で、同文中で梅棹忠夫氏の文章をそのまま引用している箇所について、段落の左右のマージンを漢字2文字分程度下げることで引用を分かりやすくしてください。また、このようにしたことをコメントとして付けてください。

9.10 PDF 形式の文書の作成

Word で作成した文書はそのままでは Word のファイル形式に対応したソフトでなければ閲覧できません。Web などでも公開されている文書には PDF と呼ばれる形式のファイルが用いられます。レポートを提出する際にも PDF 形式での提出を求められることが少なくありません。Word で作成した文書を PDF 形式にするには以下の手順で行います。

- [ファイル]メニューを選び、
[名前をつけて保存]を選択します。そしてファイルの種類として PDF を指定して保存してください。

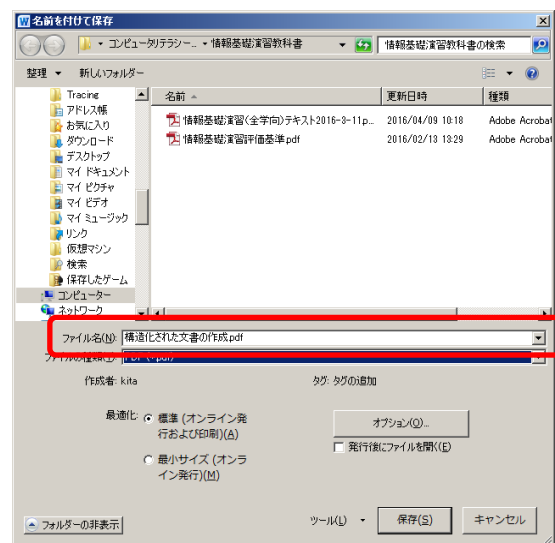


図 67 PDF 形式での保存

注意 Windows ではファイルの形式に応じた拡張子が用いられます。Word のファイル(doc や docx などの拡張子がつけられています) の拡張子をエクスプローラーで pdf に変えるとアイコンが PDF のリーダーなどに変わりますが、ファイルの中身は変わらず、中身は Word 用のファイル形式のままです。このような操作を行うとファイルを読みだすためのアプリケーションソフトウェアが分からなくなるので行ってはいけません。

参考文献

以下の書籍[9][10]は論文などの構造のある文書についての Word の使

い方を解説しています。書籍[8]は文字の配置や強調などについてレポートの資料作りの点から解説しています。

- [8] 高橋佑磨, 片山なつ: よい資料を作るためのレイアウトのルール 伝わるデザインの基本, 増補改訂版, 技術評論社 (2016)
- [9] 西上原裕明: Word で作る長文ドキュメント, 技術評論社(2011)
- [10] 相澤裕介: 論文・レポート作成に使う Word 2013 活用法, カットシステム (2013)
- [11] 科学技術情報流通技術基準 http://sti.jst.go.jp/sist/menu_purpose/index.html (2016 年 10 月 26 日アクセス)

10. 文章への図表の挿入

10.1 学習の目的

Word を用いて学術的な文章に図表を挿入する際の基本として以下のことを学びます。

- 図や表を挿入する際の番号付けや参照のルールを知ります。
- Word での番号付きの図表の説明の挿入と番号の本文での引用の方法を知ります。
- Word での図表そのものの挿入の方法を知ります。
- Excel で作成した表やグラフ，PowerPoint で作成した図などを Word の文章に挿入する際に複数の方法があることを知ります。

10.2 学術的文章での図表番号，タイトル，説明の付け方

10.2.1 番号やタイトル，説明の位置

学術的な文章での図や表は通常，以下のように扱います。⁵

- 図表には番号を振り，タイトルや説明を付けます。
 - 図の番号などは図の下側に付けることが通例です。
 - 表の番号などは表の上側に付けることが通例です。

10.2.2 本文での参照

文章に挿入した図や表は必ず本文で参照します。参照は「図 1」，「表 2」など番号で引用します。

⁵ 理工系の領域で一般に行われる方法ですが，学問領域での慣習や論文誌などの発表対象での要項に沿う必要があります。

10.2.3 出典の明記

図や表はそれ自体を他の文献から引用したり，あるいは公表されているデータをグラフにプロットしたりして作成することが多くなります。引用した図，表については出典を，データについては出典と，さらにプロットは著者自身が行ったことなどを明記します。

10.3 Word での図表番号の付与と相互参照

10.3.1 Word での図表番号の挿入

図表番号を挿入するには「参考資料」のメニューを選び，リボン「図表」から「図表番号の挿入」を選びます。

10.3.2 Word での図表番号の参照

本文中で図表番号を参照するには「参考資料」のメニューを選びリボン「図表」から「相互参照」を選びます。「参照する項目」として「図」や「表」を選び，「相互参照の文字列」については「番号とラベルのみ」とすればよいでしょう。あとは参照先から挿入したいものを選びます。

10.4 Word の文章への図表の挿入

図表を挿入するには「挿入」メニューで現れるリボンから選びます。

ブロック図などは「挿入」メニューのリボンから「図形」を選んで作成します。図は画像ファイルの挿入に使います。PowerPoint で作成した図や Excel で作成した表，グラフの挿入については以下を参照してください。

図表の周りの文章の回り込みについては，図や表を選んで右クリックして現れるメニューで調整します。

10.5 Word と Excel, PowerPoint の連携

10.5.1 データの形式と変換, コピーとリンク

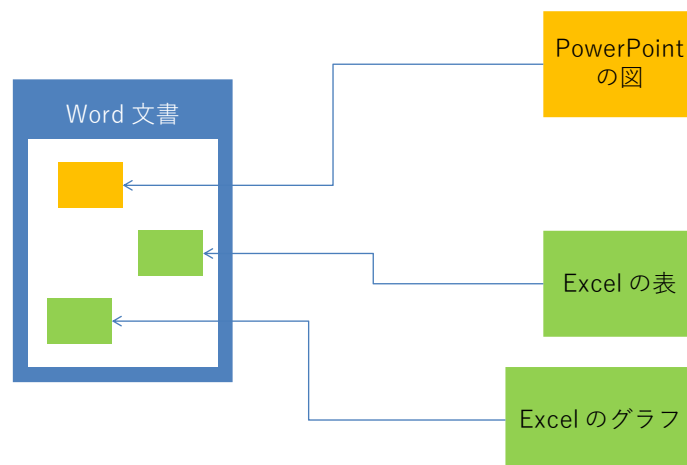


図 68 PowerPoint や Excel の情報の利用

Windows ではコピーしたり張り付けたりするデータのことを「オブジェクト」と言います。オブジェクトとしてさまざまなデータの形式があります。Excel や Power point で編集できるオブジェクトのほか、Word 内で扱えるオブジェクト、写真と同様の画像として扱うオブジェクトなどがあります。

オブジェクトをコピーして貼り付ける場合、

- もとのオブジェクトのまま扱うことや、
- 画像などの他の形式に変換して扱うこと

があります。

また、

- オブジェクトをコピーした上で貼り付ける場合と、
- もとのデータへのリンクとして扱う場合

があります。前者はコピーしてしまっているので、貼り付けたオブジェクトの編集はもとのオブジェクトと独立に行えます。後者はもとのオブジェクトにリンクしているので、もとのオブジェクトの変更が反映されます。

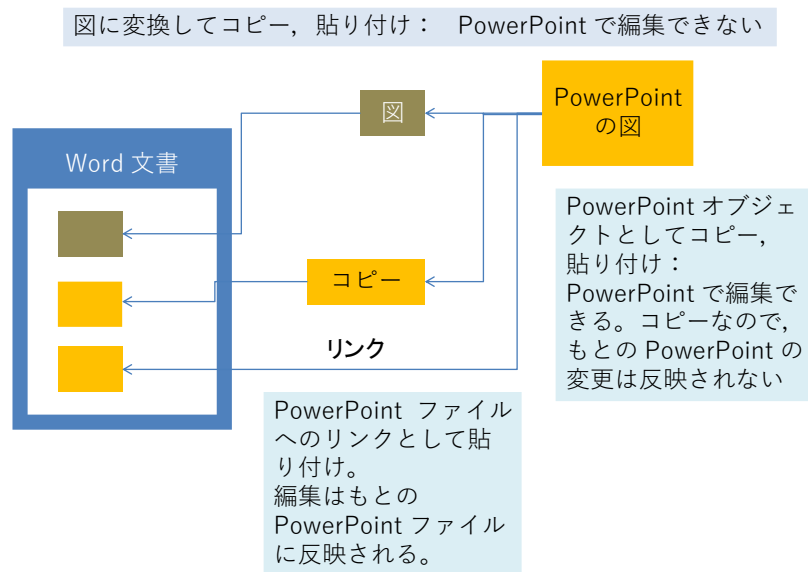


図 69 貼り付けるオブジェクトの形式と得失

10.6 PowerPoint の図の挿入

PowerPoint で作った図を挿入するには，PowerPoint の標準表示で左側に現れるスライドのウィンドウから，挿入したいスライドを選んでコピーします。次に挿入先の文書を編集している Word で

- スライドを図として挿入したい場合は，そのまま Word のカーソルがある位置で「貼り付け」をします。
- PowerPoint で編集できるオブジェクトとして貼り付ける場合は，ホームメニューのリボンの「クリップボード」から「貼り付け」の下の方の▼を展開し，「形式を選択して貼り付け」を選んで「Microsoft PowerPoint Slide オブジェクト」を選びます。
- このダイアログで「リンク貼り付け」を選ぶともとの PowerPoint ファイルへのリンクとして貼り付けます。

余白の調整は貼り付けたオブジェクトを選択し，右ボタンで現れるダイアログから「トリミング」の記号をクリックすることで行います。

10 文章への図表の挿入

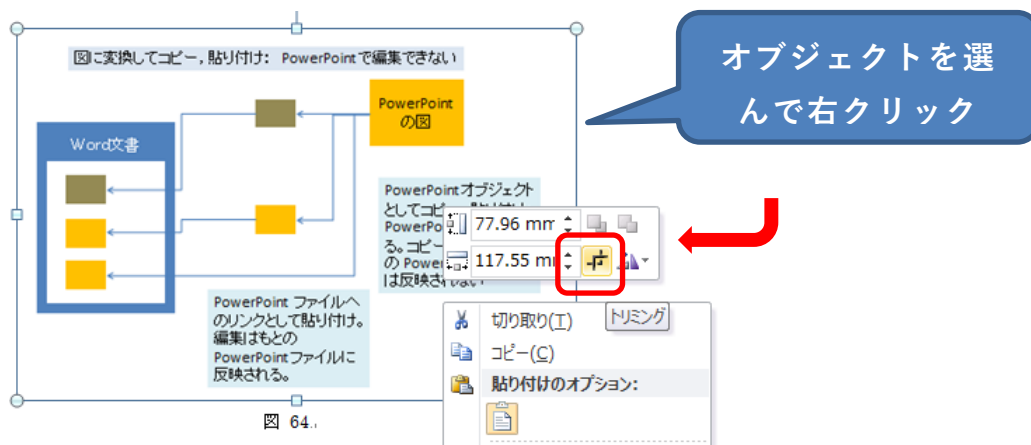


図 70 トリミングによる余白の調整

10.7 Excel の表やグラフの挿入

Excel の表やグラフも PowerPoint 同様、

- Excel で貼り付けたい部分をコピーし、
- Word で貼り付けます。
- 形式を選択して貼り付けることで Excel で編集可能なオブジェクトやもとのファイルへのリンクとして貼り付けることが可能になります。

11. 文書での数式の扱い

11.1 学習の目的

- 学術的な文章での数式の扱いについて学びます。
- 数式特有の変数などでの字体の選択について学びます。
- Word での数式の挿入方法について学びます。

11.2 表現の手段としての数式

高等学校でも数学や理科で数式を多く使ってきたと思いますが、これまでは主に数式は与えられて、それを計算したり、証明したりすることが多かったのではないのでしょうか。主体的にレポートを作成するにあたっては、**数式は「解く」ためのものである前に、数量的な関係を効果的に表現する手段**だという視点で図や表などの利用と同じように考えましょう。

数式を記述する際には使用されている変数などを正しく説明することが求められます。例えば

一定の速度で移動している物体の移動時間と移動距離の関係は

$$l = vt$$

であらわされる。

と書いただけではどの変数が何を表すか分かりません。次のように変数 l, v, t について本文中で説明する必要があります。

一定の速度で移動している物体の移動時間 t と移動距離 l の関係は

$$l = vt$$

であらわされる。ここで v は移動速度である。

11.3 変数は傾いている

数学など数式が書かれた教科書を見てみましょう。例えば2次関数について

$$f(x) = ax^2 + bx + c$$

と書かれていたりします。言われないと気づかないかもしれませんが、 x や y などの変数が x や y といったように傾いた文字（イタリック体，斜体）で書かれています。

これは斜体で書かれた文字は「量を表す変数」であること，すなわち，実際にはいろいろな値を取る得るものであることを示しています。数式には斜体で書くものと傾けずに立体（ローマン体）で書くものがあります。一般的には以下のようなルールに従っています。

- 量を表す変数は斜体で書く。
- \sin や \cos など定義の定まった関数や定数は傾けずに立体で書く。
- 単位を表す m や kg など傾けずに立体で書く。

11.4 Word での数式の書き方

Word で斜体や上付き文字，下付き文字を多用する数式を書くためには「数式ツール」を使います。

- 数式ツールの起動は以下のいずれかの方法で行います。
 - 「挿入」タブで「数式」を選ぶ。
 - Alt, Shift キーを押しながら $=$ を押す。
- 変数など斜体にするには以下のように操作します。
 - 数式ツールの「標準テキスト」を2回クリックする。
 - Ctrl-i （Ctrl キーを押しながら i をおす）を入力する。

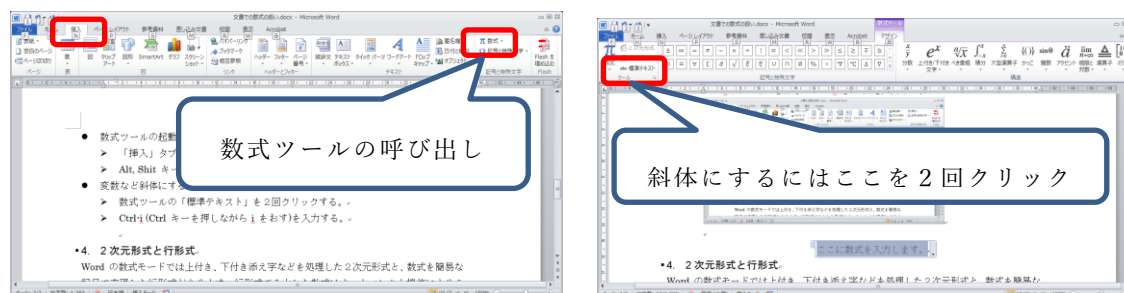


図 71 数式の挿入

11.5 2次元形式と行形式

Word の数式モードでは上付き、下付き添え字などを処理した 2 次元形式と、数式を簡易な記号で表現した行形式があります。行形式で入力した数式はオートコレクト機能により 2 次元形式に変換されます。たとえば数式ツールのなかで

$$y = ax_1^2 + bx_1x_2 + cx_2^2$$

と入力すると `_` や `^` はその後の文字をそれぞれ下付き、上付き文字として処理して 2 次元形式で

$$y = ax_1^2 + bx_1x_2 + cx_2^2$$

と表現されます¹。

¹ 数式を含む文書を作成するツールとして LaTeX が有名ですが、その記法にそった記法が採用されています。

12. プレゼンテーションについて考えよう

12.1 学習の目的

- プレゼンテーションの目的や留意事項について学びます
- スライド作成の基本的な事項を学びます
- プレゼンテーションの作法について学びます
- 聞き手として望まれることを学びます

12.2 プレゼンテーションの目的と聞き手

演習 32. プレゼンテーションは何のために行うのでしょうか？

演習 33. プレゼンテーションに際して聞き手についてどのようなことを考えればいいのでしょうか。

- 聞き手はどのような人たちでしょうか
- 話し手が聞き手に期待することは何でしょうか

12.3 プレゼンテーションでの留意事項

12.3.1 時間を共有するということ

プレゼンテーションには与えられた時間があり、それは決して多くありません。プレゼンテーションでは聞き手に（強制的に）時間を使ってもらいます。多くの聞き手に時間を有効に使ってもらうことは発表者の責務です。

演習 34. 与えられた時間内に効果的にプレゼンテーションをするためには何をすればいいのでしょうか。

12.3.2 限られた時間で伝えられることの限界

プレゼンテーションやそのためのスライドは時間的な制約の中でコンパクトに情報を伝えることに重きが置かれます。スライドはしっかり書かれた文章に比べ、情報量（文字数）が相当に少ないこと[12]には注意しなければなりません。複雑なことをしっかり伝えるためにはレポートなどの文章として表現してください。

12.4 スライド作成の基本

1) 枚数や文字の大きさの目安

スライドで使用する文字は読みやすい大きさを考えて決定する必要がありますが、**目安として10行程度**と考えます。ニュース解説などのテレビ番組でフリップが用いられるのを見ると相当に行数が少ないことが分かります。

プレゼンテーションに必要な時間の目安として、10行程度のスライドを作ると**スライド1枚を話すのに1分程度は必要**だと考えてください。もちろん、スライドに盛り込む内容の量によってその説明に要する時間は異なります、実際には練習などを通じて時間を見積もる必要がありますが、上記の見積もりで概略を想定します。

2) スライドの構成

発表時間を考慮してスライドに何を入れるかは以下の基準で考えてください。

1. 必要なものは入れて必ず話す
2. 不要なものは入れない
3. 論理を飛躍させない
4. 前から順に聞けば分かるようにする
5. 聞き手が何を理解できるか考える
6. 事実や他人の主張と自分（たち）の主張を明確に分ける

うまく話せないときには構成を見直します。

1. はプレゼンテーションを行う理由から明らかですが、限られた時間を聞き手に使ってもらうことから 2. も求められます。時間が限られていることから **1. については内容を重要なことに絞る**ことも必要になります。学術的な内容を扱う際は、3. や 6. は必須の要求です。また、内容が専門的になることが多いので 5. は聞き手への配慮として求められます。学術的な内容は、論理的な構成が重要で、なおかつ複雑な事項を話すことも多いのですが、「話す」ことが中心になりますので、内容を前後させない配慮として 4. が求められます。

3) 箇条書きについて考える

スライドは箇条書きを多用します。以下の点で考えましょう

- **同質なものが並んでいるか。** 箇条書きとする項目は並列して並ぶものですから、何らかの意味で同質なものが並んでいることが求められます。
- **順序に意味のあるのか，ないのか。** 作業の手順などは順序を変えると意味を失いません。このほか，重要度の順に並べるなど，順序に配慮することが求められます。
 - 順序に意味がなくても番号を振ると質疑をしやすいなど，よい点もあります。
 - 出典等を引用する場合は，もとの順序を守ることが求められます。発表者自身の論考のために順序を変える場合はそのことを明示する必要があります。
- **階層的に並べるべきものもある。** 例えば，この節で用いている箇条書きでも前の事項は順序に関する補足事項なので階層的に並べています。
- **箇条書きの項目数。** 箇条書きの項目数が少ないと，箇条書きにしている理由が分かりませんし，多すぎると理解が困難になります。発表者自身が項目数を決定している場合は次のことを考えます：
 - 項目数が少ないときには，考察が足りないことも多いので再考する。
 - 項目数が多すぎるときは，項目の整理が不十分なことも多いので再考する。

4) 表現の工夫

数値データはグラフ化する，数値そのものが重要な時に表にする。グラフのデータ点に数値を併記するという手法も有効です。

演習 35. 以下の2枚のスライドは同じデータを表現しています。その得失を考えてください。

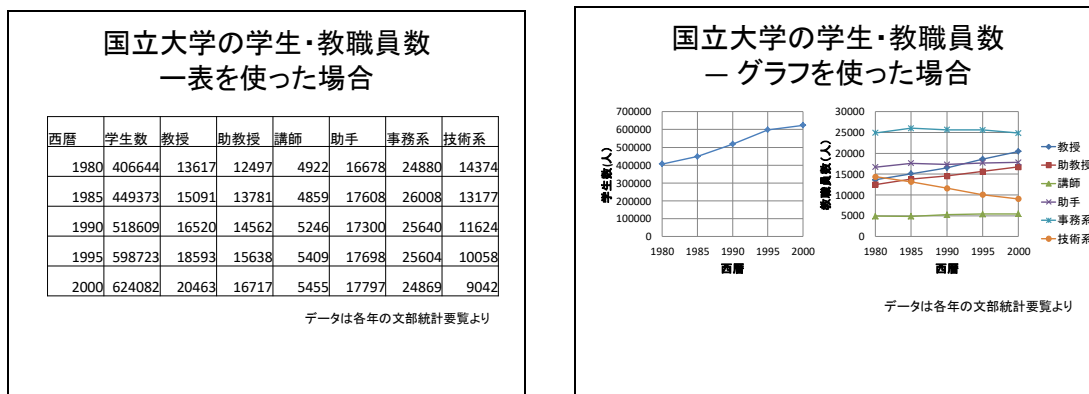


図 72 表とグラフの利用

スライドのなかでの強調

スライドで**一部を強調する**にはいくつかの方法があります。

- 文字の**大きさ**を変える
- 文字の**字体**を変える
- 文字の**色**を変える

多くの箇所を同時に強調することは難しいです。**最も重要な箇所だけを強調する**ようにします。

図 73 スライドのなかでの強調

- **強調** スライド中の文の一部を強調するには、
 - 図 73 に示すように**大きさ**，**字体**，**色**などを変えることで行います。
 - スライド全体で**統一的な強調方法**を使います。
 - 多くの箇所を強調すると相手に伝わりにくくなります。そのスライドで**最も重要なものを選んで強調**してください。
 - 単色のスライドの中で特定の所だけ別の色で文字を描くことは強調表現としては効果的ですが，**多くの色に意味を持たせて使うと，聞き手にとっては理解が困難**になります。

- **アニメーションの利用** スライドのアニメーションは見た目が面白いのでつい使いたくなってしまう。しかしながら、聞き手にとって本当に必要でなければ発表される内容に集中できないことも少なくありません。また、質疑などでスライドを行き来する際に手早く目的のスライドにたどり着けない、ということもあります。アニメーションは
 - 動的な内容を示すことや、
 - 相手に見せる内容を集中させたいときなど**真に必要な場所で使う**ようにします。
- **図や写真、ビデオの利用** 発表内容に具体性を持たせるのに効果的ですが、聞き手に与える効果が強いために不適切な図や写真の利用には注意すべきです。また、他者の権利に関わることが多いので以下の点を注意してください：
 - **著作者の権利への配慮**：イラストや写真を含め引用は著作権法に従ってください。スライドとして使用することと、それを印刷して配布すること、さらにはサーバ上にアップロードして他者が利用可能とすることは著作権法上の取り扱いが異なりますので留意が必要です。
 - **写真などに写っている人の権利への配慮**：人の肖像については「肖像権」として保護することが求められます。

12.4.2 パワーポイントのさまざまな機能の活用

1) アウトラインの活用

- スライドの構成にアウトライン（図 74）を活用します。
 - 文章として容易に書き込めます。
 - TAB キーや Shift + TAB キーでレベルを調整できます。
 - スライドの順番の入れ替えも容易です。

12 プレゼンテーションについて考えよう

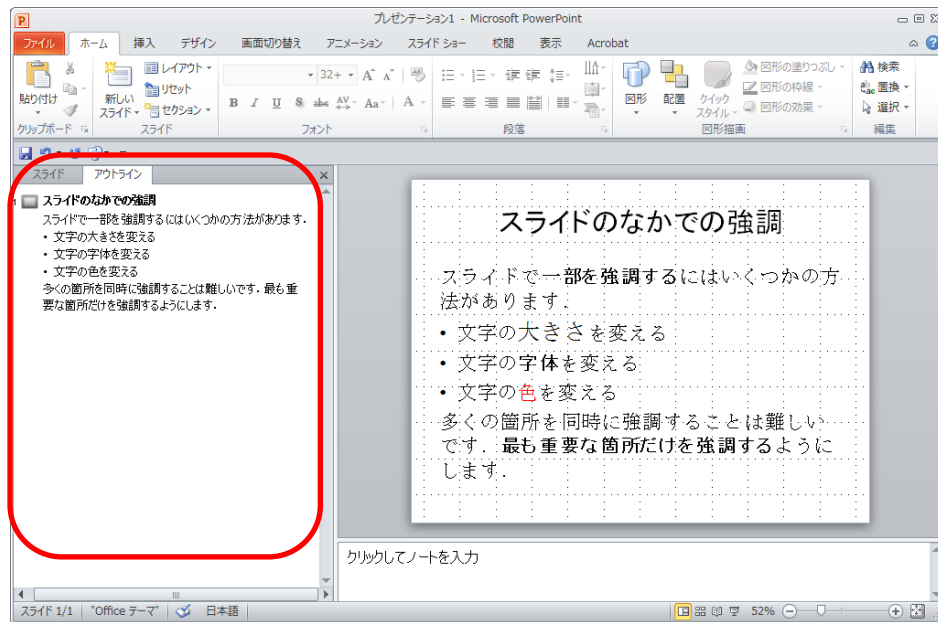


図 74 アウトラインの活用

2) スライド一覧の活用

- スライドの順番の入れ替えも容易です。
- 質疑などですばやくスライドを選ぶことができます。

12.5 発表の作法

1) 十分な準備

発表は**他人の時間を消費する行為であることを認識し**、十分な準備をしてください。

- **事前練習**：事前に練習，計時，改善します。**発表時間は厳守**します。
- **質疑への準備と対応**：質問を想定し，質疑に対する準備もしておきます。発表には含まなかった資料なども用意します。
- **発表に必要な機材**：P Cを持参するのか，発表会場で準備されるのかを確認します。P Cを持参する場合は，プロジェクタへの接続方法が VGA なのか HDMI なのかを確認します。Mac など**アダプタ**が必要なものは忘れないようにします。また，発表にビデオなどを含む場合，音声の再生をどのように行うかもチェックが必要です。
- **ポインタ**：レーザーポインタや指し棒がつかえるのかどうかを確認します。
- **配布資料**：スライドの縮小刷りなど配布資料を準備する必要はないでしょうか。

- **筆記具**：質疑でのメモをとる用意をしましょう。
- **P C の操作への習熟**：ノート P C をプロジェクタに接続するテストは必ず事前に行います。留意点として以下が挙げられます。
 - **ノート P C の画面とプロジェクタ出力のモード**：ノート P C のディスプレイとプロジェクタに同じ画面を出すモードと、別の画面を出すモードがあります。このことを理解したうえで切り替えのための操作などを練習してください。
 - **プロジェクタの信号選択**：プロジェクタはさまざまな入力を選択できるようになっています。ノート P C からの出力を選んでいるかどうかを確認ください。
 - **プロジェクタとノート P C の相性**：ノート P C からの出力はさまざまな画面の解像度で行われます。通常、プロジェクタはこれに自動で対応しますが、場合によっては対応できないことがあります。ノート P C の画面出力の解像度を調整するなどで対応します。
- **トラブルへの備え**：機材がうまく動かない場合に備えてスライドのファイルを USB メモリにコピーして持参するなども有効です。

2) 聞き手に向き合ったプレゼン

手元やスクリーンばかりを見て話す方がおられますが、これではなかなか内容を相手に伝えることはできません。聞き手へのアイコンタクトをしながら、はっきりと聞こえる声で話すことが重要です。

3) 質疑での対応

質疑に適切に対応することは発表と同様に重要なことです。プレゼンテーションは内容を相手に理解してもらうだけでなく、聞き手の意見などを聞くことも目的としているからです。

- 質問、コメントはその場でメモを取るようにします。
- 回答する前に相手の質問をどう理解したかを確認することは良い方法です。
- 意味の分からない質問を受けることがあります。これに対しては自分なりの理解でよいか質問内容を確認するようにします。
- できていないこと、考えていなかったことは取り繕わずに正直に話しましょう。

12.6 グループでのプレゼンテーション

授業ではグループで調査などを行い、その成果を発表することがあります。

A,B,C,D さん 4 人が共同で発表することを想定してみましょう。

よくあるパターンは以下のようなものです：

- 自分が作ったスライド（1/4 ずつ）を入れ替わり 4 人で話す。

演習 36. 聞き手の立場にたってみて、これは効果的と言えるでしょうか？

演習 37. 4 人で共同してよいプレゼンをするほかの方法はあるでしょうか？

12.7 聞き手としての態度

プレゼンテーションについては聞き手としてのスキルを高めることも重要です。

能動的に聞くためのスキルを身につけましょう。

- 批判的（否定的ではない）態度：ほんとうにそうなのか、論理的なのか、証拠はあるのかなどの観点で発表を聞きます。
- 質問への準備：批判的な態度で話を聞けば、確認したいこと、より深く理解したいことなど出てくるはずです。すべてを聞き終わったあとで質問を考えることは難しいので質問するつもりでメモをとるようにします。

12.8 プレゼンテーションの評価

もしプレゼンテーションを評価する、評価される、としたらどういう点を評価すべきでしょうか？以下のような評価の観点が考えられます。

- プレゼンテーションの内容
 - 興味ある、有用な内容であるかどうか
 - 聴衆を考えて理解できる内容であるかどうか
 - 内容が論理的に構成され、十分な論拠があるかどうか
 - 事実や他者の意見と発表者の意見を区別しているか
- プレゼンテーションの方法
 - 与えられた時間を適切に使っているか
 - スライドの構成が適切かどうか
 - しっかり話せているかどうか

- 図や表などを効果的に使っているか
- 視覚的効果は適切に使えているか
- 討論
 - 質疑に適切に答え、建設的な議論が行えているか

12.9 ポスター発表について

12.9.1 ポスター発表とは

研究などを発表する形式はスライドを用いて講演形式で行うもののほか、A0 や A1 サイズなどの大きな紙で作ったポスターを掲示し、その前に立ってポスターに書かれた内容について発表する形式も用いられます。講演形式の発表に比べ、並行して多くの発表を行うことができ、聴衆も興味のある発表を自由を選んで聴くことができます。

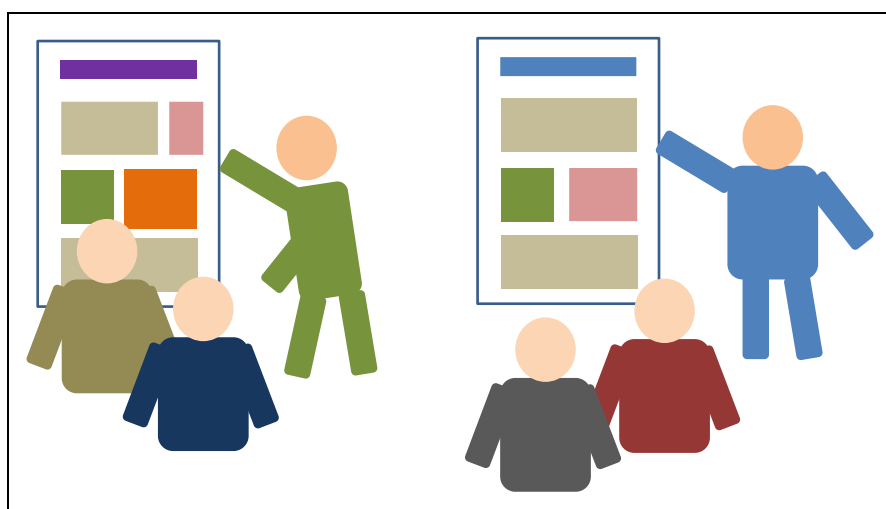
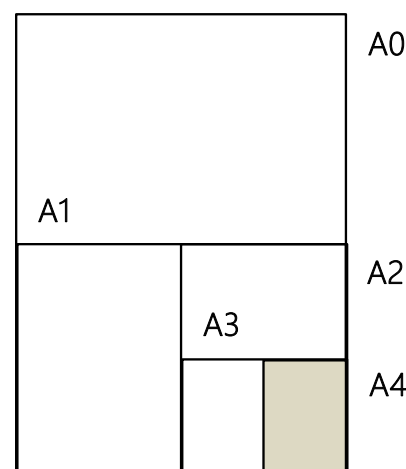


図 75 ポスター発表

12.9.2 ポスターの作成

ポスターの作成にあたっては以下のことに留意して作成します。

- ポスターを掲示する掲示板の大きさは発表会によって指定されていることが多いですが、A0, A1 などの大きさのポスターを用意することが多いです。A 系列は図 76 のように数字が 1 つ大きくなると半分の大きさになります。A1 だと A4 用紙の 8 枚分、A0 だと 16 枚分になります。
- パワーポイントでポスターを用意する場合は、ページのレイアウトで用紙サイズや向きを設定して作成します。
- 文字やグラフなどの大きさはポスターでの発表で聴衆の位置から読めるように設定します。
- ポスターは上部に発表のタイトルや発表者を分かりやすく表示します。
- 数分程度で発表することを想定して、文章やグラフなどを割り付けてゆきます。
- 大判プリンタでの印刷はかなりの費用が発生します。A4 や A3 サイズで印刷して、十分校正してから大きな紙に印刷するようにします。



12.9.3 ポスターでの発表

ポスターでの発表では以下の点に留意します。

- 数分程度で全体を発表できるように準備します。
- 発表会からの指示にしたがって掲示の方法（画鋏など掲示に必要なものを持参する必要があるのかどうか、など）を確認して準備します。
- ポスターを携行するための筒などを用意します。
- ポスター発表終了時の撤去方法も確認しておきます。
- より詳しく説明するための資料なども準備しておくといいいでしょう。
- ポスターの前でポスターを読んでいる聴衆には、「数分程度で発表しましょうか」と尋ねるようにします。
- 特定に人を相手に話すことが多いので、相手の理解を確認しながら発表します。

- 質問への回答などは講演形式での発表と同じで，メモをとったりしながら適切に対応します。

参考文献

スライドでのプレゼンテーションについてより詳しく知りたい場合は[13][14]を，ポスターについては[15]が参考になります。レイアウトやフォントの使い方については[16]が参考になります。

- [12] Edward Tufte: The Cognitive Style of PowerPoint: Pitching Out Corrupts Within, In Beautiful Evidence, Graphic Press LLC (2006)
- [13] 宮野公樹：学生・研究者のための使える！PowerPoint スライドデザイン 伝わるプレゼン 1つの原理と3つの技術，化学同人 (2009)
- [14] 宮野公樹：研究発表のためのスライドデザイン 「わかりやすいスライド」作りのルール，ブルーバックス B-1813，講談社 (2013)
- [15] 宮野公樹：学生・研究者のための伝わる！学会ポスターのデザイン術 ポスター発表を成功に導くプレゼン手法，化学同人 (2011)
- [16] 高橋佑磨，片山なつ：よい資料を作るためのレイアウトのルール 伝わるデザインの基本，増補改訂版，技術評論社 (2016)

13. プログラミングの基礎

ここではプログラムとプログラミングの基礎について学びます。この資料は演習に先立ってプログラミングの概要を把握するため、またさらに演習では言及されないであろう発展的なトピックについて学習するために利用することを想定しています。

13.1 学習の準備

プログラミングの学習では、資料を読むだけでなく、実践してみることが重要です。そこで本資料の最後でプログラミングのためのツールの設定方法とその使い方をごく簡単に紹介します。

ところでプログラミングの演習に利用する道具立てはいろいろと考えられます。学習方法もさまざまでしょう。この資料で紹介するのはそのうちの一つに過ぎません。この資料の対象となる「情報基礎演習」の科目は一つではなく、科目によって演習で使うツールや演習の進め方は少なからず異なることでしょう。そこでみなさんの受講する科目では、この資料で紹介するものとは異なるツールを利用する可能性が少なくありません。この点に注意して下さい。

13.2 学習の目的

「情報基礎演習」の仕上げとしてプログラミングを学習するのには次のような理由が挙げられます。

- プログラミングで（既存のツールでは対応できない）オーダーメイドの作業の自動化が可能になる
- プログラミングはコンピュータでの作業において同じことを何度も何度も繰り返す（退屈で）機械的な作業を避けるための手段となる
- プログラミングは面白い！（本当に！）

本来プログラミングの学習には時間をかけてじっくり取り組むのが望ましいところです。残念ながら本演習ではプログラミングの主要なトピックをすべてカバーするほどの時間はありません。しかし短い時間の中でもプログラミングの威力、面白

さ（あるいは難しさ）を体験することはできるでしょう。この演習がプログラミングを本格的に学習するきっかけになることを期待しています。またプログラミングを学ぶことは、コンピュータをより深く理解するために役立つことでしょう。

13.3 プログラムとプログラミングの概要

最初にプログラムとプログラミングについて、まずそもそもそれが何なのかということから、プログラミングを学習することの意義と学習に対する心構え、またプログラムがどのようにコンピュータで使われるのかについて説明します。

- ソフトウェア（プログラム）とその役割
- プログラミングとプログラミング言語
- プログラミングの学習
- プログラム実行の仕組み
- プログラムに潜む危険性

13.3.1 ソフトウェア（プログラム）とその役割

コンピュータでさまざまなデータを処理することは、レシピを見ながら料理することになぞらえることができます。データ処理と調理を関係づけてみると次の表 9 のようになります。

- コンピュータによるデータ処理≈レシピに従った調理
- ソフトウェア=作業マニュアル
- さまざまなソフトウェア（マニュアル）→さまざまな処理

表 9 コンピュータでのデータ処理とレシピを使った調理

データ処理		調理
元データ	処理されるモノ	材料
ハードウェア	作業する主体	調理者
ソフトウェア (プログラム)	作業手順の情報	レシピ
結果のデータ	成果物	完成した料理

調理者はレシピの指示に従って料理の材料を順に加工（切る、まぜる、焼くなど）して料理を完成させます。調理者は淡々とレシピに従えば、たとえ料理を完成させ

るまでのイメージが頭に入らなくても、悩む（考える）ことなく料理を作ることができます。レシピは（想定されるスキルがある人であれば）誰でも料理を作ることができるように書かれた調理マニュアルといえます。

さてすでに学習したようにコンピュータはハードウェアとソフトウェア（プログラム）で構成されています。

ハードウェアは実体としてのコンピュータを構成している機械です。調理でいえば作業をする調理者にあたります。ハードウェアにはデータの加工、保存、転送、つまりデータ処理を実現する多種多様な機能が備わっています。これらは調理者が持っているスキルに相当します。ただハードウェアがいろいろな機能をもっているとはいっても、機械であるハードウェアはそれだけで自律的に動作するものではありません。調理者もスキルをもっていたとしても料理の手順が頭に入っていないければ、材料から料理は作れないでしょう。

ハードウェアはいわば非常に複雑な電卓のようなものとも言えます。電卓には計算の機能は組み込まれています。しかし電卓が存在するだけで計算が独りでに進むことはありません。電卓で計算を行うには適切な順序で数値データを入力して演算機能を働かせる必要があります。人が数字や演算のボタンを押してその手順を逐一指示する必要があります。コンピュータのハードウェアでもデータ処理を実行するには、電卓と同様にさまざまな処理機能を適切な順序で駆動する必要があります。ただしコンピュータでは利用者が処理手順をすべて一々指示しているわけではありません（もしそうだとすれば人間が操作するスピードでしかデータ処理が進みません！）。

コンピュータで高速なデータ処理を実現するためにハードウェアとともに不可欠な要素がソフトウェア（プログラム）です。調理において、レシピが材料から料理を完成させる手順をステップごとに並べたマニュアルといえるように、プログラムはデータ処理の手順を記述したコンピュータ用の作業マニュアルといえます。そこにはコンピュータがどんなデータを受け取って、どんな処理をどんな順序で施して、最後にどんな形で結果を示すかがまとめられています。

調理においては、（丁寧に書かれた）レシピが用意されていれば、調理者の頭に料理の完成までのイメージが入っていなくても、レシピの各ステップをこなしていくことで自然に料理ができあがります。プログラムに従ったハードウェアでのデータ処理もこれと似たようなものと考えられます。プログラムには（ハードウェアで実行可能な）データ処理のステップが順に記述されています。ハードウェアにはプログラムのステップを順に読み取って（コンピュータが考えることなく）その指示通りに動作するメカニズムが組み込まれています。データ処理の開始を告げるのは利用者の役目です（調理では調理者が自分のタイミングで自ら作業を開始するでし

う)。メニューの項目を選択したり、ボタンを押したりすることでコンピュータに対して処理の開始を指示します。そうすればあとはプログラム（マニュアル）の通りに（ハードウェアが動作するスピードで）処理が自動的に実行されるわけです。

調理において、一人の調理者が（スキルをもっていれば）いろいろなレシピ（調理マニュアル）に従っていろいろな料理を作ることができるように、さまざまなプログラム（作業マニュアル）をハードウェアに格納しておくことで、同一のハードウェアでさまざまなデータ処理ができるようになっているわけです。なおレシピがあっても調理者にスキルがなければ料理が作れないように、プログラムを書いて実現できる処理はハードウェアに備わっている機能で実行できることに限られます。

13.3.2 プログラミングとプログラミング言語

- プログラミング=プログラムの作成
- プログラミング言語=プログラム記述のための人工言語
- さまざまなプログラミング言語がある
- プログラムの記述には厳密性が求められる

さてソフトウェア（プログラム）はどうやって用意されるのでしょうか。コンピュータを購入すると、たいてい各種のソフトウェアが最初から組み込まれていて、（ちょっとした準備をすれば）すぐにコンピュータを使い始めることができます。

（設計済みの）ハードウェアの部品は人手によって作られるのではなく機械によって作られます。それではソフトウェアも機械で自動生成するのでしょうか。（残念ながら...??）そうではありません。ソフトウェア（プログラム）は人手で作られます。プログラムを作成することをプログラミングといいます。

すでに説明した通り、プログラムはマニュアル（手順書）のようなもので、プログラミングとはマニュアルを書くことを意味します。それではプログラムはどのように書くのでしょうか。処理の手順を日本語や英語で書くのでしょうか。

プログラミングには専用に作られた人工言語を用います。プログラミングに用いる言語を総称してプログラミング言語と呼びます。プログラミング言語はただ一つではありません。世界に数多くの（人間の）言語があるように、いくつものプログラミング言語が作られています。また（人間の）言語がその特徴によって分類できるようにプログラミング言語にもさまざまなタイプのものがあります。ここでメジャーなプログラミング言語をいくつか挙げてみます（表 10）。

表 10 プログラミング言語の例

C	C++	C#	Haskell
Java	JavaScript	Lisp	MATLAB
Perl	PHP	Python	R
Ruby	Scratch	VisualBasic	

なお情報科学，情報工学の分野では，人間の言語のことを（人工の）プログラミング言語と対比して自然言語といいます．

1) プログラミング言語における厳密性

自然言語と同様にプログラミング言語には（言語ごとに）文法が定められています．プログラムは文法に従って記述しなければなりません．そんなことは当然！と思うことでしょう．しかしちょっと待ってください．我々の自然言語を思い出してみてください．もちろん自然言語でも文法はルールです．しかしそのルールはある程度は寛容なもので，表現が多少曖昧であったり，間違いがあっても許される（理解できる）ことが少なくありません．一方で「プログラムを文法に従って記述する」というとき，それは文字通りルールは絶対であることを意味します．つまりプログラムの記述においてはたった一つの間違いも許されません．コンピュータが間違いを適切に解釈し直したり，行間を読んだりするようなことはまったく期待できません（そんな高等な機能は用意されていません！）．自分が書きやすいからといって，自分が書きたいようにプログラミング言語の文法を（ちょっとだけ）アレンジして書いてみても，それは全く受け入れられません（それぐらい分かるだろうって？，ブブー，残念でした）．ただしプログラミング言語には全く自由がないというわけではありません．ある程度自分のスタイルで選択し調整する余地もあります．しかしそれは文法で許容される範囲に限定されます．やはり文法を逸脱することはできません．このようにプログラムにおいては厳密性が要求されます．

13.3.3 プログラミングの学習

- プログラミング→コンピュータを使いこなすスキル
- プログラミングの学習＝言語，データ表現，処理手順の学習

プログラミングのスキルを身に付ければ，オーダーメイドの仕事をプログラムで記述して，コンピュータに実行させることができます．プログラミングのスキルを習得するとは，いわばコンピュータを手足のように使いこなすスキルを身に付けることです．

プログラムを記述できるようになるためにはプログラミング言語を学ぶ必要があります。プログラミング言語は文法で規定されます。それなら文法を一通り学習すれば、すぐにプログラマ（プログラムを書く人）になれるのでしょうか。残念ながらそうではありません。プログラミング言語は「言語」です。英語を使いこなすために、まず英語の文法は習うとして、文法が分かったら英語が自由自在...なんてことがない（ふつうはそうですよね）のと同じで、プログラムを書くのにプログラミング言語の文法の学習は必要ですが、明らかにそれだけでは十分ではありません。

プログラミング言語を使いこなすには使い続けるしかありません（自然言語もそうですよね）。習うより慣れろです。いろいろなプログラムを読んでみて、書いてみて、言語を使えば使うほど頭の中に言語のスキルが蓄積されて、はじめは教科書を片手にトットツとしか使えなかった言語が徐々に自分のものになって、そのうちストレスなく言語が操れるようになっていきます。

さてプログラムを書くには、プログラミング言語を操れるだけでは不十分で、目的となる処理を実現する方法を見出すための素養が必要です。もう少し具体的には、処理するデータをどのように表現して、どういう手順でデータをどのように処理すれば目的を達成できるのかという問題を解決するための能力が必要です。このときとにかく目的が達成できれば（問題が解ければ）それで十分という場合もあるでしょう。しかし一般に同一の目的を達成するための処理の実現方法は一通りではありませんし、処理の方法によって効率が変わってくる場合もあります。効率のよいプログラムとそうでないプログラムでは処理時間に大きな差が生まれることも少なくありません。

データの表現方法やデータ処理の手順は自分で独自に考えることもできます。学習の過程では自分で工夫して、いろいろ試してみることも重要です。その一方で情報科学・情報工学の分野では、これまでにさまざまなデータ表現や処理方法が考案されてきています。プログラミングの学習においてはそれらを学ぶことも重要です。

以上のようなことから、プログラミングのスキルを（実用的な意味で）習得するには相当のコスト（学習時間）がかかることが分かるでしょう。しかし一度習得してしまえば、簡単に忘れてしまうことはありません。

さてプログラミングのスキルは、独自のツールを作るだけでなく、コンピュータでの作業を効率化するのにも役立ちます。コンピュータで同じような作業を延々と繰り返すことになるとき、プログラムが書けるなら、プログラムを書いた方がよいと考えることもできるでしょう。このようなスキルはコンピュータを扱う上で大きな力になりえます。

1) 失敗を恐れない

さて誰しも新しいことに挑戦するときには失敗をするものです。プログラミングの学習過程でも失敗はつきものでしょう¹。作成したプログラムに文法的な間違いがある場合はプログラムを実行（の準備を）しようとしたときに自動的に検出されます（そのときに出力される「エラーメッセージ」をヒントにして誤りを修正することになります）。プログラムが文法上正しくてもうまく動作しない場合もあります。しかしそのようなプログラムを実行したとしてもコンピュータが壊れることはまずありません²。学習段階では大いに失敗しても構いません。むしろ「失敗は成功のもと」と言われる通り、失敗から多くのことが学べるでしょう。

13.3.4 プログラム実行の仕組み — プログラミング言語とハードウェアの狭間にある機械語

- プログラミング言語＝人間のための言語
- プログラミング言語のプログラム→機械語に翻訳して実行

さて次に Python というプログラミング言語で記述した（ごく短い）プログラムの例を示します。

¹ 十分に学習した後でも、きっと失敗はなくならないでしょう...;-P

² 機械を制御するプログラムに間違いがある場合は機械を壊してしまう危険性があります。しかし学習段階ではそのようなプログラムに触れることは少ないでしょう。

リスト 1 Python プログラムの例

```
n = int(input("n? "))

fact = 1
for j in range(1,n+1):
    fact = fact * j

print('{0}!={1}'.format(n, fact))
```

Python を知らなければ，このプログラムを理解することは難しいでしょう（いまこのプログラムを理解する必要は全くありません）．しかしそれでも，このプログラムには英単語や数式らしきものが見られますし，Python について学習すれば読んで理解することは期待できそうです（そう思えますよね?）．

さてプログラムはプログラミング言語で記述します（Python であれ何であれ）．コンピュータは与えられたプログラムの指示に従ってデータを処理します．それでは具体的にコンピュータはプログラムをどのように解釈して指示を実行しているというのでしょうか．

コンピュータでデータ処理を行う実体はハードウェアであることを思い出してください．ハードウェアは単なる機械で要するに電子回路の集積物です．それではそのような機械がどのように言語の記述に従って処理を実行するというのでしょうか．プログラミング言語を「賢い」コンピュータが学習するのでしょうか（一体どうやって??）．

プログラミング言語で記述されたプログラムと機械であるハードウェアの間には明らかに大きなギャップが存在します．このギャップを埋めるのが機械語です．プログラミング言語で記述されたプログラムはそのままハードウェアで解釈されるわけではありません．ハードウェアで行われている処理とは，突き詰めればコンピュータ内のさまざまな装置に格納されているさまざまなデータの電子回路による加工や転送です．電子回路はメモリ装置に格納されたデータ（物理的な状態で表現されている 0,1 のビット列）を処理装置に適宜送り込むことで駆動されます．0,1 のデータによって，さまざまな回路のスイッチが ON/OFF されるわけです．どんなビット列のデータがハードウェアのどの回路をどう駆動するのかは機械的な仕組みとして定められています（そのようにハードウェアが組み立てられています）．

さまざまな「ハードウェアでの処理を駆動するビット列」によって定められるのが「機械語」です．例えばメモリのある場所に別の場所からデータを送り込む処理

を駆動する機械語の命令（ビット列）やあるレジスタ（特殊なメモリ）のデータを別のレジスタのデータに加算する処理を駆動する機械語の命令（ビット列）などが定められています。そのような機械語の命令（ビット列）を適宜並べることでハードウェアの処理を駆動する「本物」のプログラムである機械語のプログラムが得られます。

次に機械語プログラムの（ほんの）一部を例として示します。これは決してデタラメに作ったものではなく、本物のプログラムから抜粋したものです。機械語のプログラムのビット列にはもちろん意味が定められています。しかしとても解読できる代物ではありません。これを見れば、プログラミング言語で書かれたプログラムの方が遥かに分かりやすそうなのが実感できますよね。

```
0000000001101100011010010110001001100011001011100111001101101111
0010111000110110000000000111001101100101011101000111010101101001
011001000000000001100101011110000110100101110100000000001100101
0111100001100101011000110111011000000000011001100110111101110000
0110010101101110000000000111001101110100011100100110111001100011
0110110101110000000000000111000001100101011100100111001001101111
011100100000000000010111110101111101101001011100110110111101100011
0011100100111001010111110111001101110011011000110110000101101110
```

図 77 機械語のイメージ

機械語によるプログラミングにおいては、ハードウェアでの処理を具体的に指示する機械語命令を順に並べてデータ処理の全工程を記述することが求められます。ハードウェアの内部構造を知らなければ機械語プログラムは書けません。またハードウェアで実行される処理の一つ一つはごく単純であるため、機械語のプログラムは長くなりがちで意味を把握するのが難しくなります。さらに機械語命令はハードウェアの回路を駆動するための 0,1 のビットの並びでしかなく、単語のように読めるものではありません。

コンピュータが開発された時点ではプログラムはすべて機械語によって記述されていました。しかし機械語のプログラムは読み書きに大変な苦勞を伴います。また機械語の記述はハードウェアの動作に直結しているため、ハードウェアが異なれば、当然機械語も別のものになります。つまりあるコンピュータ用に書いた機械語プログラムはハードウェアが異なる別のコンピュータでは動作しません。

以上のように機械語によるプログラミングは決して効率的ではありません。そこ

でまずプログラムが楽に読み書きできるようにするために、機械語を英単語や数字に直訳して記述できるようにしたアセンブリ言語がコンピュータが現れてから早い段階で開発されました。それと同時に「アセンブリ言語を機械語に翻訳する」機械語プログラムであるアセンブラも開発されました。

アセンブリ言語は機械語よりも遥かに扱い易いものですが、それでも機械語を直訳しただけで機械語の欠点は解消できたとは言えません。そこでハードウェアに依存せず、さらに読み書きしやすいプログラミング言語および「プログラミング言語で記述したプログラムを機械語に翻訳する」プログラム¹が開発されました。ちなみに世界初のプログラミング言語は FORTRAN で 1957 年に発表されています²。

以上のようにコンピュータが実行しているのは（ハードウェアで機械的に解釈できる）機械語のプログラムで、プログラミング言語で書かれたプログラムは機械語に翻訳されて実行されるわけです。

Great Idea! 自動プログラミング

プログラミング言語で書かれたプログラムを機械語に翻訳する作業自身をコンピュータにさせることは「自動プログラミング」とも呼ばれます。「プログラムを作ること」自身を「プログラム」で行わせることで、複雑なプログラムの設計・開発・保守が可能になりました。

13.3.5 プログラムに潜む危険性

さてコンピュータはプログラムで指示されたことは忠実にこなします。そこで間違ったことを指示したとしてもそのままその通りに実行します。またどんなデータが与えられてもプログラムで受け入れ可能であれば処理します。コンピュータは「おっと、このプログラムは何だかおかしいぞ...」とか「このデータは怪しいな...」と考えたりはしません。

間違いが含まれるプログラムを実行した場合に起きうる問題として、まず処理の途中でプログラムが異常終了（abort）することが挙げられます。逆に処理が延々とつづいて終わらなくなってしまうこともあります。別の問題として、処理は完了しても、その処理方法が間違っている場合があります。そのことにより誤った結

¹ 翻訳プログラムには大きく分けてコンパイラとインタプリタがあります。コンパイラでは機械語のプログラムを生成します。我々は生成されたプログラムを実行することになります。一方インタプリタの場合はプログラミング言語のプログラムを読み取りつつインタプリタに組み込まれた機械語の命令を実行します（機械語のプログラムは生成しません）。なおプログラムの実行スタイルは他にもあります。

² IBM100 FORTRAN The Pioneering Programming Language
<http://www-03.ibm.com/ibm/history/ibm100/us/en/icons/fortran/> (Last access: 2016-08-29)

論が導かれ、実害が発生することもあります。たとえば人工衛星に搭載されたプログラムで数値データの正負を誤って記述したため、コントロールがおかしくなって、衛星を破棄せざるを得なくなったという問題が知られています¹。

またさらにプログラムで想定されている（通常の）利用方法において全く問題がないとしても、プログラムに（見落とされている）間違い、つまり欠陥がある場合、想定できてなかったパターンのデータをプログラムが受け付けてしまうことで、問題が発生する場合があります。マルウェア（コンピュータウイルスなど悪意の目的で使われるソフトウェア）は、そのような欠陥を悪用しています。

プログラムに含まれる間違いを専門用語でバグ（bug）とよび、プログラムを修正してバグを取り除くことをデバグ（debug）といいます。セキュリティ上の問題となる欠陥（バグ）は特に脆弱性（vulnerability）とよばれます。

さまざまなシステムがコンピュータ制御されている現在の社会において安全・安心に生活をするためには、バグのないプログラムを作ることが求められます。しかしプログラムを作るのはあくまでも（間違いを犯すことのある）人間です。一般に大規模で複雑なプログラムになるほどバグを完全になくすのは難しくなります。

13.4 変数によるモデル化

ここではプログラムの不可欠な構成要素として、変数について説明します。プログラムにおける変数とは何か、どう使われるのか、どういう効果があるのか、また変数を使うことの意義を説明します。

13.4.1 例題 — コラッツの予想

さてここで例題として、数学でよく知られている問題である「コラッツの予想」（Collatz Problem）を紹介します。

コラッツの予想

任意の正の整数 n から始めて、偶数なら 2 で割る、奇数なら 3 倍して 1 を加えることを繰り返すといつかは必ず 1 になる。

たとえば $n=3$ についてコラッツの予想が正しいことは次のように確かめられます。

$$3 \rightarrow 10 \rightarrow 5 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1$$

¹ X 線天文衛星「ひとみ」の異常に関する小委員会の検証結果について
http://www.jaxa.jp/press/2016/06/20160614_hitomi_j.html (Last access:2016-08-29)

この問題は一目ごく単純ですし、実際にいろいろな数で試してみても予想は正しように思えます。しかし数学的には正しいことが証明できていません。つまりコラッツの予想は未解決の問題です。

数学の証明問題としてコラッツの予想に取り組むことは至難の業でしょう。しかし具体的に整数 n を指定して、予想が成り立つかどうか試してみることはできます。次にそのようなテストを実行してみるプログラムをプログラミング言語 Python で示します。

リスト 2 コラッツの予想を試すプログラム

行番号	プログラム	解説
1	<code>n = int(input("n? "))</code>	キーボードから n の値を得る
2	<code>while n != 1:</code>	$n \neq 1$ である限り 2～7 行目の処理を繰り返す
3	<code> print(n,end = " -> ")</code>	途中経過の表示(現在の n の値)
4	<code> if n % 2 == 0:</code>	n が偶数(2 で割って余りが 0)のときは
5	<code> n = n // 2</code>	→ n を 2 で割った値を再び n とする
6	<code> else:</code>	そうでないとき(n が奇数のとき)は
7	<code> n = 3*n+1</code>	→ n を 3 倍して 1 を加えた値を再び n とする
8	<code>print("1 OK")</code>	ここに至った時点で $n=1$ のはず

このプログラムでは n として正の整数が指定されることを期待して、その n について予想が正しいかどうかを具体的に確かめます。計算過程での n の値の変化を順に表示していった、予想が実際に正しいのであれば最後に「OK」と表示してプログラムは終了します¹。

以下ではこのプログラムでの変数の必要性を説明します。また一般に変数を利用する意義を説明していきます。

13.4.2 変数とその必要性

- プログラムのデータは主に変数で扱う
- 変数＝データと名前を対応づける仕組み、データのモデル

¹ $n > 1$ からスタートして処理の途中で n が最初の値に戻ったとすれば、コラッツの予想が間違っていることが分かります。しかしこのプログラムではそれを確かめるようにはしていません。プログラムの中の「//」は Python での(整数の間での)商を計算する演算です。

さて例題のプログラムはさまざまな n の値についてコラッツの予想が正しいかどうかをテストできるようになっています。プログラムでは n の値は指定されていません。つまりプログラムの記述においては n の値は不定で、実行するときに初めて n の値が定まるようになっています。また n が偶数なら 2 で割る、奇数なら 3 倍して 1 を加えるという計算の結果得られる値を再び n に指定して処理を行っています。

このようにプログラムにおいて変化するデータはそのデータを象徴する名前によって表現されます。実際にプログラムが動作するときには名前にデータの値を当てはめて処理が行われます。名前とデータを対応付けて扱う仕組みを変数といいます。これは数学で扱う変数と同様のもので、プログラムでデータを表すモデルであるといえます。

13.4.3 変数の意義

- プログラム実行における変数の意義
 - プログラムの外部のデータの処理
 - プログラムの汎用化
 - 計算結果の再利用
- プログラムを書くときの変数の意義
 - 名前でデータに意味づけ
 - データとデータの関係の明示
 - プログラムの修正を容易にする

1) プログラムの実行における変数の意義

プログラムで変数を使うことの意味として、すでにコラッツの予想の例で見たようにプログラムに記述されていないデータを取り込んで処理することが挙げられます（そのようなデータを予めプログラムに記述しておくことはできません）。

また変数にはさまざまな値を当てはめられるため、同一のプログラムで（同種の）さまざまな値に対する処理の結果を得ることができます。つまりプログラムに汎用性をもたせることができます。数学の関数と同じで、変数を使うことで処理をモデル化（計算方法を名前を使って一般的に表現）できるわけです。

変数を使えば処理の効率を上げることも可能です。いま仮に、ある関数 $f(x)$ について $f(1000)$ の計算に 100 秒かかるとしましょう。このとき次の計算を考えます。

$$f(10000) \times f(10000) \times f(10000)$$

この処理では $f(10000)$ を 3 回計算するため、計算には全体でおよそ 300 秒かかることが予想されます（乗算はたいていほぼ一瞬で終わります）。このとき $f(10000)$ は同じ値に決まっているからといって、2 回目以降の計算が（自動的に）省略されたりはしません。コンピュータはプログラムに指示された通りにしか処理をしません。

$f(10000)$ とは $f(x)$ の $x=10000$ のときの計算を実行することであり、プログラムにそれを計算せよと書かれていれば（以前の処理を覚えているわけでもなく）、それに従って忠実に処理を行います¹。

さてここで次の式によって数学のように変数 y に $f(10000)$ の値を代入できるものとします。

$$y = f(10000)$$

このとき次のように変数を使って処理すれば $f(10000)$ を計算するのは 1 回だけです。そこで処理時間はほぼ 100 秒に短縮できるでしょう。

$$\begin{aligned} y &= f(10000) \\ y \times y \times y \end{aligned}$$

このように変数に途中結果を代入して再利用することで処理を効率化できます。なおプログラムにおいて同じ結果を得る処理を何度も実行する（あるいは書く）ことは無駄と考えられます。プログラムで「無駄に処理を繰り返さないようにすべし」という教訓を DRY (Don't Repeat Yourself) の原則といいます。

2) プログラムを書くときの変数の意義

さてプログラムにおける変数の威力は、実行時のみに発揮されるわけではありません。プログラミングにおいても変数は必須です。

データの値は無個性で同じ値があちこちにあったらそれらは区別できません。変数によってデータを表現すれば、その名前でデータに意味づけできます。また変数を組み合わせて処理を記述することで、どんなデータからどんなデータが得られるのか、データとデータの間係を明示することもできます。

たとえば次の (a), (b) を比べてみてください。print は画面にデータを表示する機能だとします。また「*」は乗算、「/」は除算を表しています。

¹賢い翻訳プログラムならこのような重複した処理を検出して重複を除くように最適化するかもしれません。

リスト 3 値のみのプログラムと変数を利用したプログラム

(a)	<code>print(100 * 12 / 2)</code>
(b)	<code>base = 100</code> <code>height = 12</code> <code>area = base * height / 2</code> <code>print(area)</code>

(a),(b)とも実行すれば同じ値が得られます。(a)の方が記述は簡単です。しかし見ただけでは一体何を計算しているのか分かりにくくなっています。一方(b)であれば、処理の意味はわかりやすいのではないのでしょうか。

なおプログラムにおいて、値が直接記述されていて意味が分かりにくいデータのことをマジックナンバーといいます。マジックナンバーは避けるべきです（自分が書いたプログラムでもマジックナンバーは、あとで意味が分からなくなってしまうかも！）。

変数を使えば、プログラムの修正も容易になります。データの値をあちこちに書いているとすると、値の変更が必要になったときには、それらをすべて適切に置き換えることになります（おっと、たまたま同じ値になっている別のデータを書き換えないように！）。データを変数で表現しておけば、変数に代入する値を置き換えるだけで作業は終わりです。

以上のように、プログラム、プログラミングにおいて、変数は欠かすことができない仕組みです。

13.5 制御構造 — 条件に基づく処理

ここでは変数とともにプログラムで本質的に必要となる制御構造について説明します。制御構造とは条件に基づいた処理のための仕組みの総称です。制御構造には「条件分岐」と「繰り返し処理」があります。とくに「繰り返し処理」によって大量のデータを処理するために何度も同じ（パターン）の処理を繰り返すことはコンピュータの真骨頂と言えます。

- 条件に応じて異なる処理を行う
- 条件に従って同じ（パターンの）処理を繰り返す
- 繰り返し処理→コンピュータの真価の発揮
- 停まらないプログラムもある

13.5.1 条件に基づく処理の必要性

先に挙げた「コラッツの予想」をテストするプログラムでは条件を考慮した処理が必要となります。n が偶数か奇数かによって計算処理が異なります。また n が 1 と等しくなるまで同じ処理を繰り返します。これも「n が等しくなるまで」という条件に従った処理です。プログラムではこのように条件に基づく処理が必須です。

さてプログラムにおいては処理が一つずつ順番に処理されていくのが基本動作になります。これはマニュアルに従った作業で記述されているステップを最初から最後まで順番に辿るのと同じようなものです。一方、条件に基づく処理では、条件によって異なる処理が実行されたり（条件によっては実行されない処理が出てきます、これはマニュアルでもよくあることです）、条件に基づいて同じ処理が何度も繰り返されたりする（プログラムの同じ範囲の処理を繰り返します）ことになります。このようにプログラムの処理の実行順序が条件によってコントロールされる仕組みを一般に制御構造といいます。

13.5.2 繰り返し処理の威力

さてコンピュータを使ってデータを処理することの大きなメリットは、何千、何万、何百万という（一連の）大量のデータの処理、あるいは延々と続く計算処理を高速にこなせることです。これはプログラムに繰り返しの制御構造を組み込むことで実現されます。同じ作業を延々と繰り返すことは人間には苦痛ですし、繰り返すうちに集中力を欠いて間違えることもあるでしょう。しかしコンピュータにはそのような問題はありません。与えられたデータがいくら多くても、疲れることなくいくらかでも同じ処理を忠実に繰り返すことができます（電力が供給されてハードウェアが動き続ける限り）。繰り返しの制御構造がこの力の源となります。

たとえば次の数を n として「コラッツの予想」をテストするプログラムを実行すると 2813 回の繰り返し後に 1 に到達することが分かります¹。

```
5293237546378456534784553489308590438902375927369027359027902375429185097690729571298679027590211819
```

このような数について自分で計算をして確かめるのは...実質的に不可能ですよ。しかしコンピュータで計算させればあっという間に答えに辿り着きます。これこそがコンピュータの力です。

¹ 繰り返しの回数を記憶させる処理を追加する必要があります。なおプログラミング言語によっては、ここで示したような「大きな」数と通常の計算で扱うような「小さな」数では異なる処理が必要になります。これはコンピュータ内でのデータの表現が「小さな」数と「大きな」数では異なるためです。

13.5.3 停止しないプログラム

さて繰り返し処理についてももう少し説明します。これまでに見てきたように繰り返し処理とは「条件が正しい限り繰り返す」処理です。ここで「ファイルの各行を順に読み取ってそれぞれに同じ処理を施す」ような処理を考えてみましょう。これは「ファイルから行が読み取れる」という条件による繰り返しで実現できます。具体的にはファイルの先頭から1行ずつ読み取って処理することを繰り返すと、いつかは最後の行に到達して、その次に「ファイルから行が読み取れる」という条件が成り立たなくなってしまう。このような処理は（ファイルがいくら長くても）いつかは終わります。

それでは全ての繰り返し処理は「いつかは終わる」ことが決まっているのでしょうか。このような問いかけがされることから想像できるように、ずっと終わらない繰り返し処理もありえます。単純には繰り返しつづけるための条件として「常に正しいこと」を指定すれば、（コンピュータが動作する限り）永遠に終わらないプログラムになることが分かるでしょう。次にそのようなプログラムの簡単な例を示します（とくに意味のあるものではありません）。

リスト 4 停まらないプログラムの例

行番号	プログラム	解説
1	<code>i = 1</code>	i の値を 1 とする
2	<code>while i > 0:</code>	i > 0 である限り 3,4 行目を繰り返す
3	<code> print(i)</code>	i の値を表示する
4	<code> i+=1</code>	i の値を 1 増やす

このプログラムでは 1,2,3,... と順に i の値を増やしながら処理を進めていきます。i > 0 である限りは処理を繰り返すことからプログラムを実行したら（理論上は）ずっと動き続けることになります¹。また先に挙げた「コラッツの予想」のプログラムは、コラッツの予想が正しくなければ、反例となる n を指定したときに（理論上は）プログラムは停止しないことになります。

なお OS の管理下で動作しているプログラムは OS によって強制的に停止させることはできます（さらに言えば電源を切ればどんなプログラムでも停止するに違いあ

¹ 実際にはコンピュータで処理できるデータ量に限界があるため停止するはずですが、停止するまでには相当な時間がかかることでしょう（筆者も実際に試してみたことはありません）。

りません！)¹。

ここで見たような停まらないプログラムが書けてしまうことは問題に思えるかもしれませんが、しかし必ず停止することが保証できる「安全な」範囲にプログラムを制限してしまうと、そうでないときに比べて、プログラムで実現できる処理も制限を受けてしまうことが分かっています。

13.6 プログラミング演習の準備

ここではプログラミングを実際に行うための方法の一つとして、本資料でも取り上げているプログラミング言語 Python でプログラムを作成、実行などするためのツールのインストールとその使い方について Windows を対象にしてごく簡単に説明します（MacOS や Linux でもおそらく同様に準備できるでしょう）²。

なおこの資料は 2016-07-15 現在のものです。この資料の公開後に Python が更新されて、以下の情報が古くなっている場合もありえますので注意して下さい。

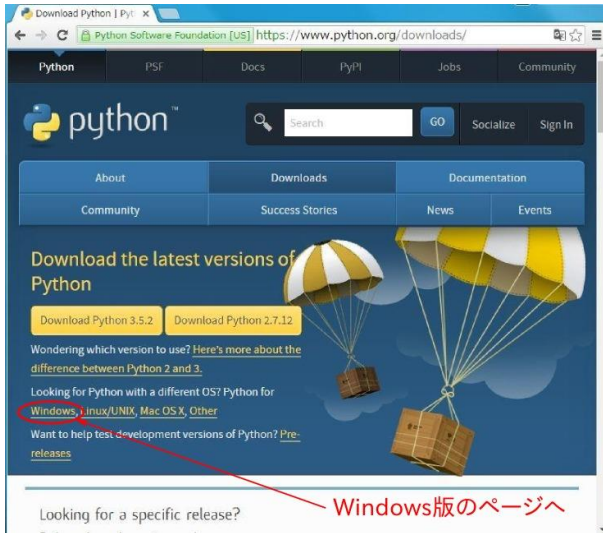
13.6.1 Python のインストール

最初に Python のパッケージをインストールします。Python の実行に必要なソフトウェアおよび Python のプログラムの作成に利用するエディタ（編集ソフト）等が一式含まれています。

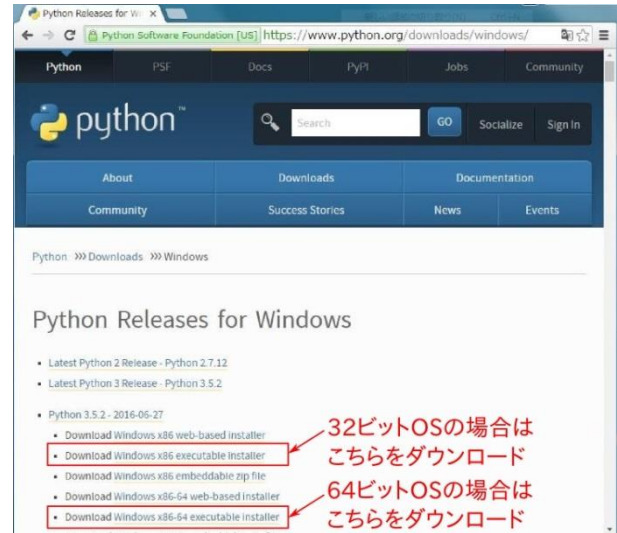
1) インストーラのダウンロード

まず適当な WWW ブラウザで Python の公式 WWW サイトのダウンロードのページ（<https://www.python.org/downloads/>）を開きます（図 78 (a)）。つづいて Windows 用のページに移動します（図 78 (b)）。開いたページからインストール用のパッケージをダウンロードします。ダウンロードするパッケージには 32 ビット OS 用と 64 ビット OS 用があります。利用している OS のバージョンによって適切に選択して下さい。バージョンは Windows の「コントロールパネル」の「システム」以下を探せば確認できます。

¹ この資料の最後に紹介する IDLE の場合、プログラムの実行中に [Shell] メニューの [Restart Shell] を選びます。
² 現行の最新版である Python3 を対象とします。一つ前のバージョンである Python2 も現状ではよく使われています。Python2 と 3 では文法に小さくない差異がありますので注意して下さい。



(a)



(b)

図 78 Python 公式 WWW サイトのダウンロードページ

2) パッケージの確認(option)

ダウンロード後、インストーラを実行する前に、パッケージが不正なものではないことを確認すると安心です。具体的にはセキュリティソフトでインストーラをスキャンすることができます。また Python のサイトに掲載されているハッシュ値 (MD5), GPG の署名を使ってパッケージの整合性を確認することもできます。ここではこれらの確認方法について詳細は説明しません。関連する情報は「Latest Python 3 Release」のページで確認してください。

3) インストールの実行

ダウンロードしたインストーラを実行するとインストールが開始されます。インストーラを起動したら、最初の画面で「Install launcher for all users」の選択を外して、「Add Python 3.5 to PATH」を選択した上で「Install Now」を選んで下さい(図 79 (a)). これでインストールが開始されて問題がなければそのままインストールが完了します(図 79 (b)). Python が「すべてのアプリ」¹に追加されているはずです。

¹ Windows 7 なら「すべてのプログラム」

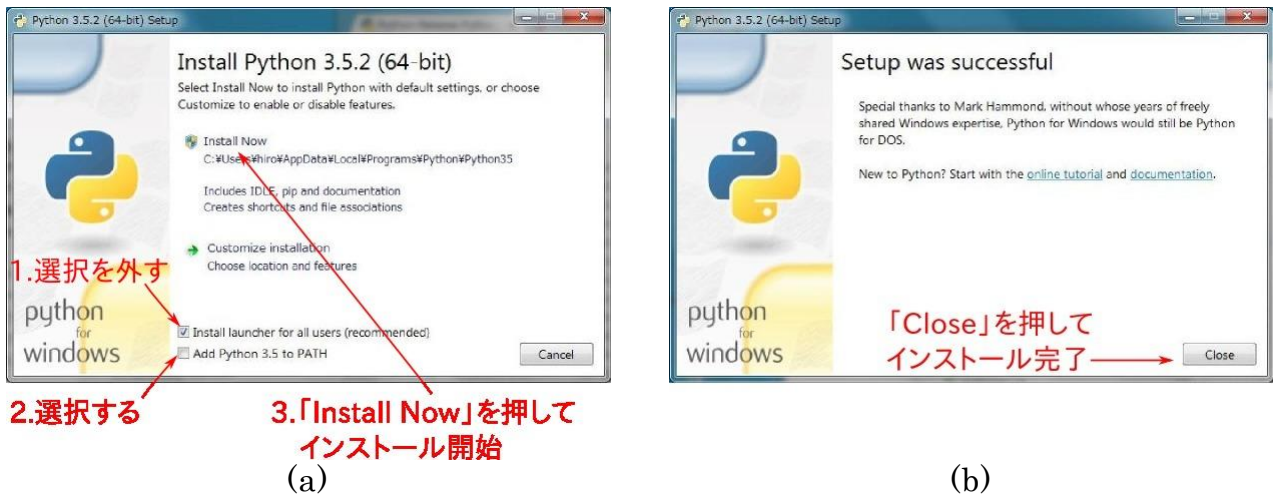


図 79 Python のインストール

13.6.2 Python を試してみる

それでは Python のプログラムを作成，実行してみましょう．インストールされた Python のメニューに「IDLE」というツールが見つかるはずです．これを起動すると次の図 80 (a)のような画面が出てきます．この画面ではシェル（Shell）というプログラムが動作しています．画面の「>>>」はプロンプトといって，シェルがデータの受け付けを行っていることを示しています．画面にプログラムを入力して[Enter]を押すと，すぐに処理が実行されます．図 80 (b)は簡単な式を評価したところです．

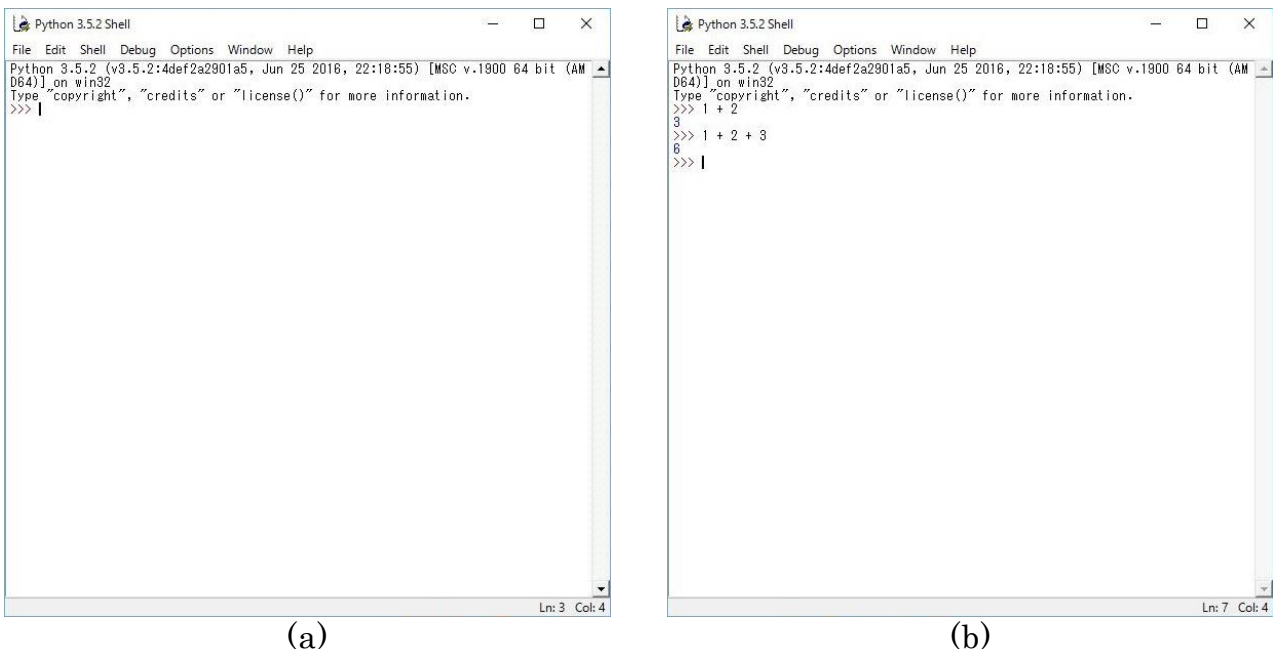


図 80 IDLE の起動画面と簡単なテスト

さて本格的なプログラムは別途ファイルに記述します．シェルのメニューの[File]

13 プログラミングの基礎

から[New File]を選んで下さい。 図 81 (a)に示す新しい画面が開きます。これがプログラムを作成・編集するツールでエディタと呼ばれます。エディタの画面にこの資料に掲載している「コラッツの予想」のプログラムを入力してみましょう。

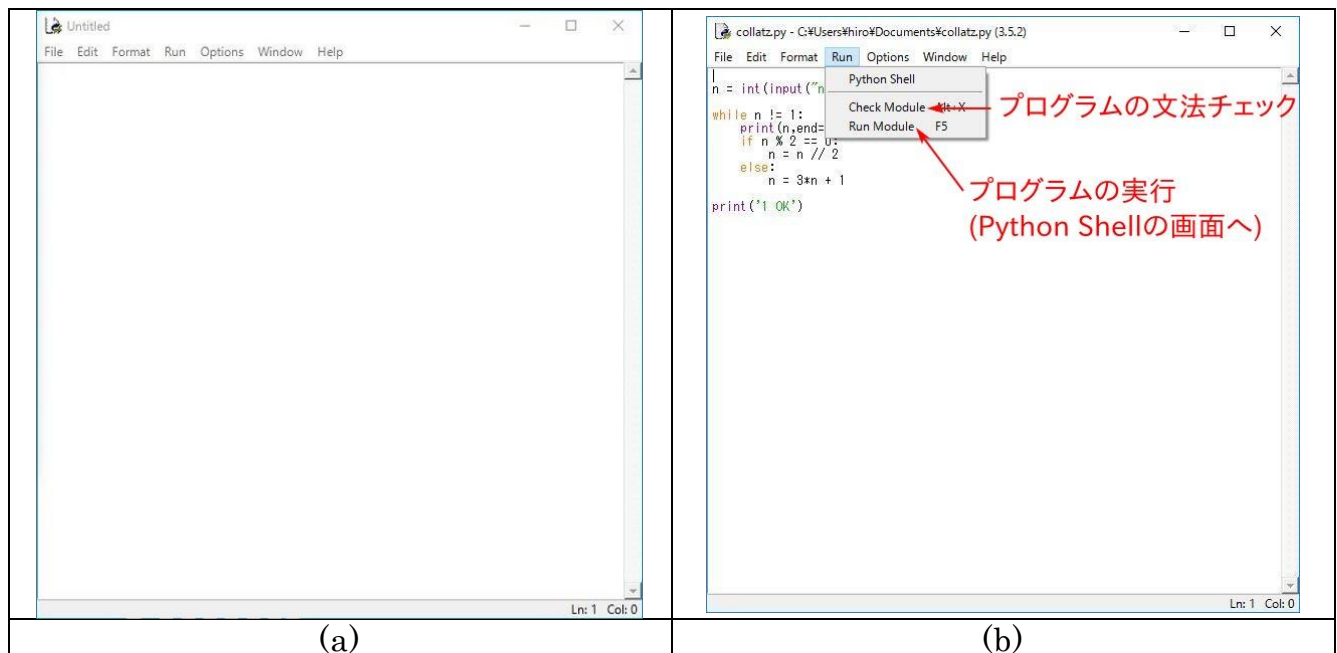


図 81 IDLE のエディタ

リスト 5 コラッツの予想を試すプログラム（再掲）

行番号	プログラム	解説
1	<code>n = int(input("n? "))</code>	キーボードから <code>n</code> の値を得る
2	<code>while n != 1:</code>	<code>n</code> ≠ 1 である限り 2～7 行目の処理を繰り返す
3	<code> print(n, end=" -> ")</code>	途中経過の表示(現在の <code>n</code> の値)
4	<code> if n % 2 == 0:</code>	<code>n</code> が偶数(2 で割って余りが 0)のときは
5	<code> n = n // 2</code>	→ <code>n</code> を 2 で割った値を再び <code>n</code> とする
6	<code> else:</code>	そうでないとき(<code>n</code> が奇数のとき)は
7	<code> n = 3*n+1</code>	→ <code>n</code> を 3 倍して 1 を加えた値を再び <code>n</code> とする
8	<code>print("1 OK")</code>	ここに至った時点で <code>n=1</code> のはず

入力したらファイルを適宜保存しておいて下さい。 つづいて図 81 (b)に示しているようにエディタの[Run] (実行) メニューの[Check Module]でプログラムの文法をチェックします。間違いが指摘された場合は修正した上で再度保存して下さい。問題がなかった場合には[Run Module]でプログラムを実行できます。このプログラムでは

13 プログラミングの基礎

実行するとすぐに「n?」と表示されて、n の値を入力することになります（図 82）。適当な 2 以上の正の整数を指定して[Enter]を押すと 計算が始まります。その後（コラッツの予想が正しければ）ほどなく処理が終了します。

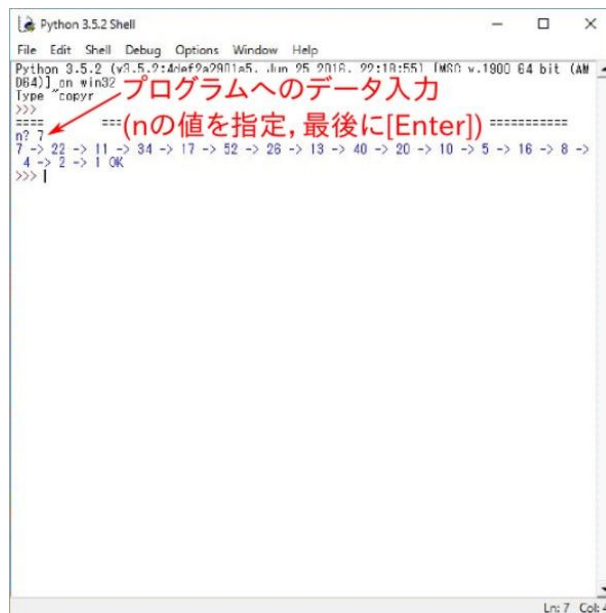


図 82 コラッツの予想のプログラムの実行

最後に次のようにプログラムの 4 行目と 8 行目を書き換えて保存した上でさきほどと同様に実行してみてください（変更部分は赤字で示しています）。

リスト 6 コラッツの予想のプログラム（データ出力形式を変更したもの）

行番号	プログラム	解説
1	<code>n = int(input("n? "))</code>	キーボードから n の値を得る
2	<code>while n != 1:</code>	n ≠ 1 である限り 2～7 行目の処理を繰り返す
3	<code>print(n)</code>	途中経過の表示(現在の n の値)
4	<code>if n % 2 == 0:</code>	n が偶数(2 で割って余りが 0)のときは
5	<code> n = n // 2</code>	→n を 2 で割った値を再び n とする
6	<code>else:</code>	そうでないとき(n が奇数のとき)は
7	<code> n = 3*n+1</code>	→n を 3 倍して 1 を加えた値を再び n とする
8	<code>print(1)</code>	ここに至った時点で n=1 のはず

今度は数値が 1 行に 1 つずつ表示されるはずです。たとえば処理が終了するまで

の過程での値の変化をグラフにして解析しようとするのであれば、このようにデータを表示した方が便利でしょう。またこのようにプログラムを書き換えてみることで、プログラムに従ってデータ処理が行われていることが実感できるのではないかと思います。

参考文献

Python(Python3)の参考書の例を示します。

- [17] 鎌田 正浩: 確かな力が身につく Python「超」入門, SB クリエイティブ (2016/3/16)
ISBN: 978-4-797384406
- [18] Guido van Rossum (著) 鴨澤 眞夫 (訳): Python チュートリアル 第3版, オライリー
ジャパン (2016/3/24) ISBN: 978-4-873117539

14. あとがき

この教科書ではアカデミックな活動での ICT 利用スキルについて学びました。1.1 節でも述べたように、この教科書で学んだことから、さらにご自身に興味や専門的な分野の必要性に応じで様々な ICT 利用スキルを主体的に獲得していくことが求められます。別表 1 に示した評価基準もこのことを意図して作成していますが、そのほかすべてを身に着ける必要もないかと思いますが、以下のようなスキルの獲得も考えられます。

- オンラインストレージなどの効果的な活用
- パーソナルコンピュータとスマートフォンと連携利用
- 英語、日本語以外の言語での文字入力
- 文献管理ソフトの利用
- データベースの構成と利活用
- ビデオの編集と利活用
- CAD や 3 次元プリンタの利用
- Web ページの作成とプログラミング
- SNS などのソーシャルメディアの効果的な利用
- プロジェクトの管理とそのためのツールの利用

これらのスキルを学んで行くための学習資源を探索したり、共に学んだり、教えあったりする仲間のネットワーク作りも重要な視点です。

コンピュータを中心とする情報通信技術の特徴は自身で新しいツールを創出できることです。本書ではコラムとして **Great Idea!** を設けて情報通信技術は誰かが素晴らしいアイデアを得て作り上げてきたものであることを紹介しました。将来、皆さんの貢献が **Great Idea!** として紹介できるなら、それは素敵なことだと思いませんか。

別表 1 評価基準

別表 情報基礎演習 評価基準

本評価基準は情報基礎演習で扱う項目について学習者の達成度を評価するための基準である。レベルはこの科目の合格レベル（C），達成目標レベル（A），在学中に身につけて欲しいレベル（S）の3段階とした。背景が白色のものは扱わない

大項目	中項目	小項目	レベルC	レベルA	レベルS	備考
			必要な技能を不十分ながら獲得できている(演習科目合格レベル)	必要な技能を基本的なレベルで獲得できている(演習科目達成目標レベ	在学中に身につけて欲しいレベル	実施の考慮事項と評価方法
学内情報サービスとネットワークの利用						
	学内情報サービス		学内で提供されている情報サービスを理解しているが一部のみ利用できる	学内で提供されている情報サービスを理解し利用できる	学内で提供されている情報サービスを効果的に利用して学習に役立てている	履修登録、学習管理システム(LMS)、学内ネットワークなどの利
	私物の P C などの学内ネットワークへの接続		VPN等大学が提供するサービスを利用して P C やスマートフォンなどを援助者の支援を得て大学のネットワークに接続できる	接続における暗号化などを理解し、VPN等大学が提供するサービスを利用して P C やスマートフォンなどを大学のネットワークに接続できる	機器の特性や利用環境に応じて複数の接続方式を利用できる	実技での評価
	電子メールを用いたフォーマルなコミュニケーション					
		適切な内容のメールの作成	件名、差出人、受信者を明示したメールを書ける	件名、差出人、受信者を明示し、適切な内容と言葉遣いのメールを書ける	Cc, Bcc, Reply-to などのフィールドを適切に利用してグループでのコミュニケーションに活用できている	メールの利用演習
		受け取ったメッセージの正しい取扱い	受け取ったメールのメッセージの適切な扱いを理解し、実践できている	同左	同左	テストなどで評価
		添付ファイル利用時の配慮	サイズが適切な添付ファイルの送受はできるがサイズの圧縮や暗号化はできない	適切なサイズ、形式の添付ファイルを送受できる。重要な内容の添付ファイルの暗号化ができる。	ファイルを送受する他の手法とその安全な利用を理解し、メールと組み合わせで適切に利用できる	メール利用演習
	授業で利用する コース管理システム などについての操作		LMSを用いたファイルの閲覧や課題の提出は行えるが教員の指示には十分に従っていない	LMSを用いたファイルの送受、課題の提出などを適切に行える	履修登録システム, LMS, 学生用メールなどを適切に連携させて利用できる	LMS(PandA)の利用で評価
	情報ネットワークを利用する上での注意		情報ネットワークを利用する上での注意事項をある程度は知っているが、実践は不十分である	情報ネットワークを利用する上での注意事項を知り、適切に実践している	情報ネットワークを利用する上での注意事項について主体的、継続的に情報を獲得し、実践を改善している	情報セキュリティ e-Learning の修了
大学における知的生産（梅棹忠夫の造語）と ICT 活用			大学での授業や課外活動などでの ICT 活用シーンを理解しているが活用は限定的である	大学での授業や課外活動などでの ICT 活用シーンを理解し、活用している	大学での授業や課外活動などで ICT を積極的に活用している	以下の各項目で評価
パーソナルコンピュータとOS、その構成と基本操作						
	自立したパソコン所有者として、コンピュータやOSについて					
		コンピュータの仕組みと O S の役割の理解	コンピュータの仕組みや O S の役割の基本的な事項をある程度知っているが、安全な設定で利用するには他者の支援を要する	コンピュータの仕組みや O S の役割の基本的な事項をっており安全な設定で利用できている	コンピュータの仕組みや O S の役割の基本的な事項を知っており、トラブル対策や性能向上などをはかることができる	テストなどで評価
		ソフトウェアのライセンスと適切な利用	ソフトウェアのライセンスをある程度理解し、問題ない程度に利用している	ソフトウェアのライセンスを理解し、適切な状況で使用している	ソフトウェアのライセンスを理解し、適切な状況で大学の所有するライセンス、オープンソースのソフトウェアなどを使用できる	同上
		P C への外部機器等の接続	P C のネットワークへの接続や、プリンタや外部記憶装置を接続には他者の支援を要するが、運用はできる	自分自身で P C をネットワークに接続したり、プリンタや外部記憶装置を接続して利用できる	有線・無線 L A N などを適切に設定して P C やプリンタなどを接続することができる	同上
	アプリケーションを利用する基盤としてのOSにまつわる操作（ファイル、デバイス（プリンタやネットワーク）、プロ		O S が管理している事項について必要な設定等を補助者の支援を得て行え	O S が管理している事項について必要な設定等が自分自身で行える	P C を効果的・効率的に利用するために O S 等の設定を行える	同上
	タッチタイピング、ショートカットキー、英文タイピングの基礎		1) タッチタイピングの指使いを理解し、ある程度実施している。 2) ショートカットキーを利用している。 3) 英文のタイピングルールを知り、実践している。 4) 適切な文字コードで入力でき	1) タッチタイピングの指使いを理解し、実施している。 2) ～4) 同左	1) 手元を見ることなくタイピングができる。 2) マウスが使えない状況でもある程度の P C の操作が可能である。 3) 同左 4) 適切な文字コードを使用するとともに必要に応じて変換できる。	タイピング速度、英文で作成した文書
情報探索と図書館利用			大学図書館等が提供している情報源について理解し、必要な情報がある程度探索できる	大学図書館等が提供している情報源について理解し、必要な情報を網羅的・系統的に探索できる	大学図書館等が提供している情報源について理解し、必要な情報を効果的、効率的に探索し、批判的に利用することができる	テストや、検索演習で評価
表計算ソフトウェアによる情報の加工						
	機械処理に適したデータの表現、入力値の制限方法、データの検査		機械処理に適した表を構成できる。	機械処理に適した表を構成できる。表記揺らぎなどを避けるため入力を制限できる	レベル A に加え、入力するデータの検査を効果的に実施できる	表計算ファイル
	データ操作：データの並べ替え（ソート）や単純／クロス集計といったデータ分析の基本		データを並べ替えて分析することができる。単純集計が行える	レベル C に加え、クロス集計が行える。	レベル A 抜加え、データ分析の必要に応じて、複合的な条件付きの集計などが行える	表計算ファイル
	表計算ソフトを用いた数値シミュレーション		与えられた差分方程式のシミュレーションが行える	パラメータを変化させたシミュレーションを行える	自らの必要に応じてシミュレーションを構成できる	同上
	データのグラフへの可視化、学術的なレポートに必要なグラフの作成		データをグラフにプロットできるが、適切な書式を選べていない	データをプロットしたグラフが作成できる。軸の書式等が適切に設定され必要な情報が盛り込まれている	グラフの形式を主張したい内容に応じて適切に選択でき、適切な書式でグラフを作成できる。	同上
レポート作成の技法とワードプロセッサの利用						
	レポートし主とした文書のワードプロセッサ等での作成上の					
		構造のある文書の作成	空白や改行の挿入など物理的表現を直接用いることで段落や見出しなどを構成できる	段落や見出しなどテンプレートの機能を用いて構成できる。 アウトラインの機能を用いて編集ができる。	レベル A に加え、必要に応じてテンプレートの調整などができる	作成した文書
		図や表の挿入	図や表を挿入した文章を作成できる。 図や表の見出し、番号を手作業で付	図表番号の挿入とその参照を自動化できている	レベル A に加え、必要に応じてフォーマットの調整ができる	同上

適切な文献の書誌情報の記述とその引用、著作物の適切な引用		適切な書誌情報を付して参考文献等を挙げるができる。 剽窃と引用の違いを理解し適切な引用が行えている。	引用箇所と参考文献について同一文書内で、自動的に相互参照する機能を使える。 参考文献を適切に管理できる。	レベルAに加え、文献管理ツールなどを活用して文献の効率的な管理と活用ができる。 分野に応じた参考文献や引用の形式を理解して、それに従って文書を作成できる。	テストならびに作成した文書	
		校閲など著作物を介したコミュニケーションとそのためのスキル	印刷した文書への校閲指示にしたがって文章を修正できる	校閲機能を用いて修正個所の明示化、コメントの付与、修正の反映などが行える	他者の書いた原稿に対して校閲を行える	校閲過程
ネットワークを利用したコラボレーション						
	ファイルやドキュメントの共有などグループワークを支えるネットワーク上でのコラボレーション技法		公開されているサービスを使ってファイルにアクセスできる	公開されているネットワーク上のサービスを利用して適切なアクセス制限を行い、著作権法を遵守したファイル共有が行える。	ネットワークなどを利用してグループで円滑な活動が行える	LMSの利用
	ソーシャルメディアなどの利用における注意		ソーシャルメディア上の情報の持つリスクを理解し倫理的な行動がとれる	ソーシャルメディア上の情報の持つリスクを理解し、適切な情報発信が行える	ソーシャルメディアを適切な利用が行え、積極的に利用できる	情報セキュリティe-Learningの修了
プレゼンテーションの技法						
	スライドの作成					
		スライドの構成	スライドの内容に一部、不十分な点はあるが概ね論理的に整っている	スライドの内容が論理的に整っている	聴衆に配慮したスライドの効果的な組み立てが行えている	発表もしくはスライド
		スライドへの図や表の挿入	スライドに図や表を挿入して利用できるがプレゼンテーションでの活用が不十分である	スライドに図や表を挿入して利用できる	図や表を効果的に利用してプレゼンテーションできる	発表もしくはスライド
		視覚的効果の利用	視覚的効果を使っているが使い方が十分に適切とは言えない	視覚的効果を適切に使っている	視覚的効果を効果的に利用してプレゼンテーションできる	発表もしくはスライド
	プレゼンテーション					
		与えられた時間でのプレゼンテーション	概ね、時間を有効に活用したプレゼンテーションが出来ている	時間を有効に活用したプレゼンテーションが行える	適切な時間配分でプレゼンテーションを効果的なものに行っている	発表
		質疑への対応	スライドを参照するのに手間取る、質疑を記録していない	スライドを参照しながら質疑に適切に答え、内容を記録している	質疑を想定した追加のスライドなどを準備して建設的な討議を行える	発表
		チームでのプレゼンテーション	プレゼンテーションを効果的なものにするためチームで共同作業がある程度できている。	プレゼンテーションを効果的なものにするためチームで共同作業できている。	チームのメンバーが欠けた時でも相互に補い合ってプレゼンテーションができる	発表と相互評価
		各種ダイアグラムやチャートなどグラフィカルな情報の表現技法	不十分ではあるがダイアグラムなど図的表現とそれを用いたプレゼンテーションが行える	ダイアグラムなど図的表現とそれを用いたプレゼンテーションが行える	図的表現にプレゼンテーションを効果的に行うための工夫を加えている	発表もしくはスライド
	プログラミングの基礎（オプショナル）		プログラミングを体験しているが主体的にプログラムは作れない	プログラミング言語の基本的な要素を理解しており、簡単なプログラムを書くことができる	自身の問題解決の道具として簡単なプログラミングを常に利用できる	プログラミング課題で評価